

1.	INTRODUCTION	1-1
1.1.	Généralités	1-1
1.1.1.	Le micro-ordinateur	1-1
1.1.2.	Les systèmes d'exploitation	1-5
1.1.3.	Le système d'exploitation PROLOGUE	1-7
1.1.4.	Conventions de notation	1-8
1.1.5.	La notion de fichier sous PROLOGUE	1-10
1.2.	Mise en œuvre de PROLOGUE	1-11
1.2.1.	Configuration du système	1-11
1.2.1.1.	Principe de la configuration	1-11
1.2.1.2.	Paramètres de configuration	1-12
1.2.1.3.	Fichiers système	1-16
1.2.2.	Chargement de PROLOGUE	1-18
1.2.3.	Mise à jour de la date et de l'heure	1-21
1.2.4.	Les unités implicites	1-21
1.2.4.1.	Définition	1-21
1.2.4.2.	Modification des unités implicites	1-22
1.2.5.	Prémarquage des supports disque	1-22
1.2.6.	Sauvegarde de vos fichiers	1-23
1.2.7.	Arrêt du système	1-26

1. INTRODUCTION

1.1. GENERALITES

Sans prétendre faire une description exhaustive d'un micro-ordinateur, il semble utile de rappeler brièvement les principaux éléments d'un micro-ordinateur.

1.1.1. Le micro-ordinateur

Un micro-ordinateur est un matériel essentiellement électronique permettant le traitement de l'information.

Le préfixe "MICRO" précise la taille et la capacité de traitement du matériel, par opposition à mini-ordinateurs et gros ordinateurs.

Il est cependant courant de trouver des grandes différences de capacités de traitement entre les très nombreux micro-ordinateurs du marché. Les configurations haut de gamme rivalisent avec les mini-ordinateurs, le bas de gamme étant à peine plus performant qu'une calculatrice de poche.

Le cœur du micro-ordinateur est constitué d'un micro-processeur, composant électronique miniature pouvant exécuter un ensemble d'instructions élémentaires qui, combinées, permettent le traitement d'informations évoluées.

Les micro-processeurs pouvant recevoir PROLOGUE sont à ce jour :

- le 8080 et ses compatibles (8085, Z80)
- le 8086 et ses compatibles (8088, 80186, 80286)

Un micro-ordinateur contient également une mémoire de lecture/écriture appelée mémoire centrale ou mémoire RAM (de l'anglais Random Access Memory), dans laquelle le micro-processeur puise les instructions et les données qu'il doit traiter et range ses résultats.

Cet ensemble est appelé l'Unité Centrale ou U.C.

De nombreux autres composants sont présents dans un micro-ordinateur, mais nous ne les détaillerons pas dans ce manuel.

Des ressources périphériques extérieures sont associées au micro-ordinateur, permettant la communication entre le micro-ordinateur et son utilisateur.

Les principales ressources périphériques d'une configuration sont :

1. La CONSOLE utilisateur composée de :

- Un écran de visualisation sur lequel sont affichés tous les messages d'informations émis par l'ordinateur.
- Un clavier, plus ou moins ressemblant à celui d'une machine à écrire, permettant d'introduire des commandes d'exploitation ou des données à traiter.

Dans une configuration multiposte ou multi-utilisateur, chaque utilisateur possède sa propre console.

La console permet donc le dialogue homme/machine nécessaire à la mise en œuvre du micro-ordinateur.

2. Les unités de DISQUE :

Les unités de DISQUE permettent le stockage et l'accès aux informations de votre exploitation, sous forme de FICHIERS.

La notion très importante de FICHIER est développée dans le chapitre 2.

Une unité de disque se compose en général d'un lecteur relié au micro-ordinateur permettant la lecture ou l'écriture du disque, lui-même contenant un support magnétique sur lequel sont enregistrées les données.

Le disque peut être FIXE, intégré de façon permanente au micro-ordinateur (INAMOVIBLE), ou bien AMOVIBLE, intégré dans une cartouche.

Dans le cas où le disque est amovible et suivant vos besoins, un autre disque peut être installé à la place du précédent et un même disque peut être transporté d'un micro-ordinateur à un autre.

Actuellement, les unités de disque les plus courantes sont :

- Les unités de disquettes composées d'un lecteur, disposées le plus souvent sur le face avant de la machine et de disquettes (ou disques souples), d'un diamètre de 5 pouces 1/4.
Les disquettes doivent être introduites correctement dans le lecteur pour pouvoir les exploiter. La capacité de ces disquettes s'étend de 300 000 caractères à un million de caractères environ.
- Les unités de disque fixe, également d'un diamètre de 5 pouces 1/4, composées d'un lecteur et d'un disque inamovible, offrant des capacités de cinq millions à plusieurs dizaines de millions de caractères.

Les combinaisons les plus courantes sont :

- Deux unités de disquettes de capacité identique, permettant une manipulation facile des informations. Dans les commandes d'exploitation de PROLOGUE, qui vous sont présentées par la suite, ces unités sont désignées par les appellations "FL0" et "FL1" (mnémoniques).
- Une unité de disquette permettant le transfert d'information à l'extérieur du micro-ordinateur et d'une unité de disque fixe offrant une capacité de stockage importante, parfois requise par l'exploitation. L'unité de disquette sera désignée sous l'appellation "FL0", l'unité de disque fixe, en général, sous l'appellation "MD0".

D'autres types d'unités de disque existent, sous d'autres combinaisons et d'autres désignations, mais les principes restent les mêmes. Reportez-vous à la documentation propre à votre micro-ordinateur.

Dans ce manuel, le terme disque désigne indifféremment une unité de disque, quel que son type : disque fixe, disquette, cartouche amovible, etc ...

3. L'unité d'impression ou IMPRIMANTE :

L'unité d'impression permet d'obtenir une trace écrite des informations : listes, tableaux, courrier, etc ...

Les caractéristiques des imprimantes varient suivant l'usage qui en est fait.

Dans les commandes de PROLOGUE, la désignation de l'imprimante se fait sous l'appellation "LO", ou "IMO" dans certains cas. Il est possible d'intégrer plusieurs imprimantes dans une configuration, désignées IM0, IM1, ..., ou IX0, IX1, ...

Les périphériques présentés ci-dessus constituent la configuration minimum la plus courante. D'autres types de périphériques existent (bande magnétique, voies de télécommunication, crayon optique, etc ...) dont l'usage est plus spécialisé et qui ne sont pas abordés dans ce manuel.

Exploiter les données d'une application consiste essentiellement à assurer le transfert de ces données entre les ressources périphériques et la mémoire centrale et à en assurer le traitement en mémoire centrale, par le micro-processeur.

La description sommaire faite ci-dessus vous permettra de mieux comprendre les notions abordées par la suite.

1.1.2. Les systèmes d'exploitation

Les travaux que vous confiez à votre micro-ordinateur se traduisent par un certain nombre d'opérations auxquelles vous n'avez normalement pas accès. Ces opérations se rapportent essentiellement à la gestion interne du système : supervision des entrées/sorties vers les mémoires de masse, imprimantes, écrans et claviers, gestion de l'espace mémoire centrale, chargement et exécution des commandes opérateur, vérifications diverses, etc ...

Un ensemble de programmes spécialisés et adaptés à votre machine prend en charge toutes ces tâches de manière permanente et transparente à l'utilisateur. Cet ensemble, qui est constitué par une réunion de "modules" spécifiques (écrits en langage machine), constituent ce qu'on appelle le SYSTEME D'EXPLOITATION du micro-ordinateur (S.E. en abrégé).

Les facilités offertes par un S.E. vous déchargent donc du souci de devoir contrôler vous-même le déroulement réel, physique des opérations, que ce soit pour l'utilisation ou le développement d'applications.

Le système d'exploitation est chargé en mémoire centrale et devient actif au démarrage de votre micro-ordinateur. Il est disponible de manière permanente, lors de l'exploitation de vos programmes.

Des programmes additionnels, dits "programmes UTILITAIRES", vous aident à exécuter des tâches à caractère répétitif ou systématique, d'intérêt général, telles que création, duplication de fichiers, édition de textes et gestion générale de votre application, etc ...

Ces UTILITAIRES sont exécutés à la demande, demande exprimée par des commandes frappées au clavier de votre console et correspondant à des programmes binaires enregistrés sur les unités de disques de votre machine.

Il vous incombe d'assurer la cohérence de votre exploitation. Par exemple, il est nécessaire d'effectuer une sauvegarde périodique des données de votre application, au moyen de l'utilitaire adapté à cette fonction.

Le système d'exploitation et les programmes utilitaires associés sont enregistrés sur un support disque qui peut être, soit une disquette, soit un disque qui sera désigné comme le disque "SYSTEME".

Ce disque système peut, soit être préalablement préparé par vous-même, soit un disque fourni déjà prêt. Le disque système est préparé au moyen d'une phase de "CONFIGURATION". Cette phase de configuration qui nécessite une connaissance plus approfondie du système, n'est exécutée en général qu'une fois, lors de l'installation du système sur votre machine. Cependant, si vous souhaitez modifier certaines caractéristiques matérielles ou logiques de votre système, il est nécessaire de reconfigurer votre système. Il existe différentes méthodes pour le faire, dont certaines sont décrites au chapitre 7 de ce manuel.

1.1.3. Le système d'exploitation PROLOGUE

PROLOGUE est un système d'exploitation MULTITACHE pour micro-ordinateurs MONOPOSTE ou MULTIPOSTE, utilisant les microprocesseurs 8 bits (8080 et compatibles) et 16 bits (8086 et compatibles).

La version de PROLOGUE, décrite dans ce manuel, est une version 16 bits, uniquement.

«MULTITACHE» signifie que plusieurs tâches peuvent être exécutées, simultanément.

Par exemple, il est possible d'exécuter en même temps une édition sur imprimante et un programme utilitaire de recopie.

L'ordinateur, n'ayant qu'un seul microprocesseur, chaque tâche est exécutée pendant un court laps de temps, successivement, donnant ainsi l'apparence complète de la simultanéité des traitements.

Ce partage du temps entre les différentes tâches est assuré par PROLOGUE d'une manière complètement transparente pour l'utilisateur.

«MULTIPOSTE» signifie que PROLOGUE peut gérer simultanément plusieurs postes utilisateurs, rattachés à la même machine. Chaque poste se compose d'un écran et d'un clavier (une console), permettant un dialogue opérateur avec PROLOGUE.

Chaque utilisateur peut donc, à volonté, lancer l'exécution de commandes correspondant, soit à des programmes utilitaires, soit à des applications particulières.

Chaque utilisateur partage, de manière concurrente, le même micro-ordinateur et ses ressources : même microprocesseur, même mémoire, même unités périphériques (disques, imprimantes, ...). C'est le rôle de PROLOGUE que d'assurer une distribution homogène et cohérente des ressources entre les différents utilisateurs du système, au mieux des performances du matériel.

1.1.4. Conventions de notation

Lors de la description des commandes opérateur, les symboles suivants sont utilisés :

- [] Les crochets contiennent une ou plusieurs options dont la présence est facultative dans la commande. Si aucune option n'est sélectionnée, une option par défaut est implicitement définie par le programme utilitaire exécuté. Les symboles "crochets" ne doivent jamais apparaître dans la commande elle-même.

Exemple :

[,LIS=LO] Permet optionnellement une édition sur l'imprimante.

- { } Les accolades indiquent un choix entre plusieurs options s'excluant mutuellement. Vous devez **OBLIGATOIREMENT** choisir l'une de ces options et une seule. Les symboles "accolades" ne doivent jamais apparaître dans la commande elle-même.

Exemple :

{ ,CO
,LO
,nomfic } L'élément référencé peut être, soit l'écran, soit l'imprimante, soit un fichier, mais obligatoirement un et un seul de ces éléments.

- <> Ces symboles encadrent la touche de validation du clavier à utiliser dans la commande. Ces symboles ne doivent jamais apparaître dans la commande elle-même.

Exemple :

<rc> Indique qu'il y a lieu d'effectuer une validation de la commande par la frappe de la touche "retour chariot".

<esc> Indique qu'il est nécessaire de valider la touche ESCape, ou sortie.

Les caractères ci-dessus sont appelés des méta-symboles, car ils servent à la description syntaxique de la commande, dans ce manuel, mais ne sont pas des éléments de la commande elle-même.

Tous les autres caractères, tels que virgule, point, etc ..., doivent apparaître tels que dans la commande concernée.

Minuscules/MAJUSCULES

Il vous est possible de frapper les caractères alphabétiques au clavier, en position majuscule ou minuscule. Toutes les commandes de PROLOGUE admettent indifféremment l'un ou l'autre et l'interprétation de la commande sera rigoureusement la même.

Cependant, dans la présentation des commandes faites dans ce manuel, certains éléments de la commande apparaissent en minuscule, d'autres en majuscule, suivant la convention ci-dessous :

- * Un élément en minuscules représente un élément variable de la commande dont la valeur effective est attribuée lors de la frappe de la commande.

Exemple :

,desfic, "desfic" peut représenter le fichier "SOURCE—S" ou "FLO.DATA-I" ou tout autre nom valide de fichier. Dans la description de la commande, "desfic" fait référence à une désignation de fichier, de manière générale.

- * Un élément en majuscules représente un mot-clé dont la valeur est fixe et dont la signification est liée au programme utilitaire lui-même.

Exemple :

,DEST=nomfic, Le mot-clé "DEST=" attribue un sens à l'élément suivant qui représente un nom de fichier.

Ce mot-clé doit donc être frappé exactement de la même façon dans les commandes où son usage est indiqué. Evidemment, la frappe peut être faite indifféremment en position majuscule ou minuscule.

1.1.5. La notion de fichier sous PROLOGUE

Un fichier est une collection d'informations enregistrées sur un support périphérique et cataloguées sous un nom unique permettant de désigner le fichier. La structure des informations, à l'intérieur d'un fichier, dépend de la nature des données qu'il contient et le système d'exploitation PROLOGUE permet d'effectuer toutes les opérations de gestion du fichier : lecture, écriture, mise à jour, modification, suppression, etc ...

Les opérations sur les fichiers représentent une part importante de l'exploitation d'un ordinateur. Leur mise en œuvre se fait, soit au niveau du système d'exploitation lui-même, à l'aide du dialogue opérateur disponible à partir de la console de chacun des utilisateurs, soit au niveau des programmes et applications que l'utilisateur mettra en œuvre lors de son exploitation.

La notion de fichier est donc une notion fondamentale d'un système d'exploitation avec laquelle vous deviendrez familier.

L'organisation des fichiers de PROLOGUE est décrite de manière détaillée dans le chapitre 2.

1.2. MISE EN ŒUVRE DU SYSTEME PROLOGUE

1.2.1. Configuration du système

Afin de s'adapter aux matériels utilisés et aux différents besoins des utilisateurs, PROLOGUE peut être configuré par l'utilisateur, lors de son installation sur le micro-ordinateur.

Il existe différents moyens pour configurer le système :

- Un moyen direct, à l'aide des utilitaires de base PATCH et EDV, rapide mais nécessitant une bonne connaissance de la structure du système.
- Un moyen plus facile, à l'aide d'un programme particulier appelé CONFIGURATEUR, qui prend en charge la configuration système à l'aide d'un dialogue opérateur directif.

Les éléments précédents sont décrits dans le chapitre 7 de ce document.

Une opération de configuration est une opération qui n'intervient qu'une fois, lors de l'installation de PROLOGUE.

Dans les paragraphes qui suivent, seule la description formelle des paramètres de configuration est faite afin d'explicitier les notions qu'ils recouvrent.

1.2.1.1. Principe de la configuration

La configuration est une opération qui consiste à construire le système PROLOGUE le mieux adapté à vos besoins, en fonction de vos critères d'exploitation.

En d'autres termes, il s'agit ici de puiser dans une réserve de modules fournis en distribution et qui sont :

- Le noyau système et le noyau environnement qui, associés, donnent le fichier SYSTEME, chargé en mémoire par la machine à l'initialisation.
- Les modules d'extension système qui apportent, chacun, des fonctionnalités supplémentaires (Séquentiel Indexé, etc ...).

- Les modules d'extensions environnement qui permettent d'intégrer des configurations périphériques particulières (imprimante, disquette 8 pouces, etc ...).

Les modules extensions se présentent sous forme de fichiers binaires indépendants qui sont installés en mémoire, lors de l'initialisation.

1.2.1.2. Paramètres de configuration

Ces paramètres permettent d'agir sur les caractéristiques du noyau système PROLOGUE. Leur valeur est enregistrée directement dans le fichier binaire SYSTÈME. Après modification de ces paramètres, ils ne deviennent actifs que lors du prochain chargement du système.

Ils se divisent en deux groupes :

- Les paramètres généraux, définissant les caractéristiques générales du système, communes à l'ensemble des utilisateurs,
- Les paramètres d'écran, spécifiques à chaque poste utilisateur et permettant de le personnaliser.

PRINCIPAUX PARAMETRES GENERAUX CONFIGURABLES :

*** Code Langue**

Définit la langue de base des messages systèmes et du dialogue des utilitaires. Les messages système sont regroupés dans un fichier source éditables, dont le nom est "SYSERF-S", si le code langue est FRANCAIS et "SYSERGB-S", si le code langue est ANGLAIS. D'autres langues peuvent être définies par l'utilisateur, par changement de son contenu. Ce fichier source est chargé en mémoire à l'initialisation du système. Les messages de dialogue des utilitaires sont inclus dans le code même de l'utilitaire et sont définis en FRANCAIS et ANGLAIS, systématiquement.

Le code langue définit la nature du séparateur des décimales, utilisé par le langage B.A.L., lors de l'affichage de nombres décimaux :

FRANCAIS	séparateur décimal = ',' (virgule)
ANGLAIS	séparateur décimal = '.' (point)
AUTRES	séparateur décimal = ',' (virgule)

* Unité Implicite SYSTEME

Définit l'unité de disque sur laquelle sera recherché l'utilitaire "SYSCONF-X" qui charge les modules d'extension, ainsi que le fichier de configuration "SYSCONF-S". Les modules d'extension seront aussi chargés à partir de cette même unité, si aucune autre unité n'est mentionnée dans leur description au sein du fichier "SYSCONF-S".

La mnémonique de l'unité Implicite Système est IS. Elle peut remplacer partout FLO, FL1, MD0, etc...

Exemple :

-> IS.CP

fait référence à l'utilitaire CP situé sur l'Implicite Système (IS).

* Nombre de postes

Définit le nombre de postes utilisateurs qui seront activés lors de l'initialisation du système. A chaque poste est associée une console composée d'un écran et d'un clavier, nécessaire pour permettre un dialogue entre l'utilisateur et le système.

Le nombre de postes varie entre 1 et 8 maximum.

* Nombre d'écrans virtuels

Définit le nombre d'écrans virtuels au total. (cf. § 6.11). Au maximum, 4 par poste.

* Caractéristique imprimante

Définit le nombre de lignes par page et le nombre de colonnes par ligne de l'imprimante courante du système. Ce paramètre n'est significatif que pour les configurations n'ayant qu'une seule imprimante et n'est pas pris en compte par certains programmes.

PRINCIPAUX PARAMETRES D'ECRAN CONFIGURABLES :

* Unité implicite PROGRAMME

Définit, pour le poste considéré, l'unité de disque sur laquelle sera recherché, par défaut, le programme binaire désigné dans la commande utilisateur.

Cette définition ne s'applique donc que dans le cas où la désignation de l'unité n'est pas explicitement indiquée dans la commande.

Exemple :

Si l'unité implicite programme du poste 0 est FL0, la commande
“—> CP,.....<rc>”

recherchera le programme “CP—X” sur l'unité FL0, alors que la commande

“FL1.CP,.....<rc>”

recherchera “CP—X” sur “FL1”.

La mnémonique de l'unité Implicite Programme est IP. Elle peut remplacer partout FL0, FL1, MD0, etc...

* Unité Implicite FICHIER

Définit, pour le poste considéré, l'unité de disque sur laquelle sera recherché, par défaut, le ou les fichiers désigné(s) dans la commande utilisateur.

Cette définition ne s'applique donc que dans le cas où la désignation de l'unité n'est pas explicitement indiquée dans la commande.

Exemple :

Si l'unité implicite fichier du poste est FL1, la commande
“—> prog,MONFIC—S,..<rc>”

recherchera le fichier “MONFIC—S” sur FL1, alors que la commande

“—> prog, FL0.MONFIC—S<rc>”

le recherchera sur FL0.

La mnémonique de l'unité Implicite Fichiers est IF. Elle peut remplacer partout FL0, FL1, MD0, etc...

Exemple :

—> IS.CP, DF, IP.MONFIC 1, IF.MONFIC 2

change CP à partir de l'Implicite Système (IS), est MONFIC 1 sur l'Implicite Programmes (IP), et l'écrit sous le nom de MONFIC 2 sur l'Implicite Fichiers (IF).

* Unité Implicite IMPRIMANTE

Définit, pour le poste considéré, l'unité implicite imprimante, choisie par défaut, pour les éditions demandées à partir de ce poste.

Exemple :

Si l'unité implicite imprimante est "IM0", la commande
"→ /,LIS=LO"
provoquera l'édition du catalogue sur l'imprimante IM0.

Il faut noter qu'il n'est, en général, pas possible de désigner explicitement l'unité imprimante, dans une commande utilisateur. Une syntaxe du genre "LIS=IM0" est en général, refusée.

Il est cependant possible de le faire, au niveau des programmes écrits en B.A.L.

* Protections d'exploitation

Ce paramètre permet d'associer spécifiquement pour le poste considéré, certaines protections d'exploitation, sous forme d'autorisation ou d'interdiction de certaines fonctionnalités. Les protections possibles sont :

1. Sauvegarde et restaurations autorisées par les commandes :

- CP,CV,...
- CP,DV,...
- CP,RV,...

2. Ecriture, en format libre, autorisée par les commandes :

- PATCH,...
- CPS<rc>

3. Disponibilité des instructions système dans le langage B.A.L. (consulter le manuel de référence du langage B.A.L.).
4. Usage du configurateur autorisé (voir chapitre 7).
5. Plus de 18 blocs et extension pour un fichier, cela signifie que la désignation des fichiers est possible suivant le format 8-3, 8 caractères pour le nom et 3 caractères pour le suffixe et qu'un fichier peut avoir plus de 18 blocs d'extension. Dans le cas d'une interdiction de cette fonctionnalité, les fichiers sont désignés suivant le format 7-1, compatible avec PROLOGUE-80 (8 bits).
6. Usage du traducteur B.A.L. et de l'option DEBUG de l'exécuteur B.A.L., autorisé.
L'interdiction de cette fonctionnalité supprime toute possibilité de développement B.A.L.

* Indicateur de commandes initiales

Permet de prendre en compte le fichier de commandes initiales, après l'initialisation du système.

A chaque poste est associé ou non un fichier différent dont le nom est "SYSTnn—C" (où nn indique le numéro de poste de 00 à 07), ce qui permet de définir des commandes initiales, indépendantes pour chacun des postes ; la syntaxe de ce fichier est identique à celle d'un fichier d'enchaînement de commandes utilisé avec l'utilitaire ASG (voir chapitre 4) et peut être créé à l'aide d'un éditeur de textes standard (éditeur EDV par exemple).

Remarque :

SYSTnn—C doit être sur l'Implicite Programme (IP).

1.2.1.3. Fichiers système

Une configuration opérationnelle du système PROLOGUE se compose des fichiers suivants :

SYSTEME	Fichier binaire contenant le noyau environnement de la machine utilisée et le noyau système. Ce fichier est installé en mémoire par le chargeur de la machine, à partir d'une unité de chargement (disquette ou disque fixe si possible).
---------	---

- SYSCONF—S** Fichier source éditable contenant la déclaration des modules extension associés au système par configuration. Chacun des modules référencés correspond à un fichier binaire de suffixe "X", présent sur l'unité implicite système. Le fichier SYSCONF—S doit être également présent sur l'unité implicite système.
- YSERF—S** Fichier source éditable, optionnel, contenant le libellé français des erreurs système. Ce fichier doit être présent sur l'unité implicite système.
- YSERGB—S** Identique à YSERF—S, mais en version anglaise, ou seconde langue.
- SYSTnn—C** Fichier source éditable, optionnel, contenant la liste des commandes initiales du poste nn. Ce fichier doit être présent sur l'unité implicite programme du poste pour lequel les commandes sont définies. Ce fichier n'est pris en compte que si l'indicateur d'activation a été correctement positionné dans le contexte du poste.
- SYSCONF—X** Utilitaire de traitement du fichier SYSCONF—S, exécuté automatiquement sur le poste 0, après chargement du fichier SYSTEME. Ce programme doit être présent sur l'unité implicite système.
- ASG—X** Utilitaire optionnel de traitement des fichiers SYSTnn—C. Il n'est utile que si des commandes initiales ont été définies pour un des postes utilisateurs. Cet utilitaire doit être présent sur l'unité implicite programme du poste concerné.
- modules—X** Tous les modules dont le nom est référencé dans le fichier SYSCONF—S doivent également être présents sur l'unité implicite système.

Se reporter au chapitre 7, pour obtenir des précisions sur la configuration et l'installation des fichiers système.

1.2.2. Chargement de PROLOGUE

Le chargement est l'opération qui consiste à installer en mémoire le système d'exploitation présent sur disquette ou disque fixe, sous la forme des fichiers système décrits ci-dessus.

Il est entendu que le système PROLOGUE que vous souhaitez utiliser a été préalablement **INSTALLE** et **CONFIGURE** en fonction de vos besoins, par vos soins ou ceux de votre distributeur.

Le chargement doit être exécuté lors de chaque mise en route du micro-ordinateur, le contenu de la mémoire étant complètement perdu lors de son arrêt. Le chargement du fichier SYSTEME est exécuté par un programme spécial appelé chargeur et intégré à la machine de façon à ce qu'il soit immédiatement exécutable après la mise sous tension. Le processus de chargement peut varier suivant les micro-ordinateurs et il y a lieu de se reporter à la documentation du Constructeur pour connaître les détails de mise en œuvre.

Cependant, le processus qui se déroule le plus couramment est le suivant :

1. Après mise sous tension de la machine, le fichier système est recherché sur la première unité de disquettes (unité 0) à une adresse précise sur la disquette (secteur physique 100 hexa).

Si le fichier système est présent à cet endroit, il est chargé en mémoire et initialisé. La suite du chargement est alors prise en charge par le noyau système lui-même, en fonction des paramètres de configuration préalablement définis.

2. Sinon, le fichier système est recherché sur la première unité de disque fixe (si elle est présente) à une adresse précise sur le disque (secteur physique 100 hexa).

Si le fichier système est présent à cet endroit, il est chargé en mémoire et initialisé. La suite du chargement est alors prise en charge par le noyau système lui-même, en fonction des paramètres de configuration préalablement définis.

Le chargement est donc réalisé sans aucune intervention utilisateur, soit à partir d'une disquette, soit à partir d'un disque fixe, en fonction de la périphérie dont est équipé le micro-ordinateur et des paramètres de configuration retenus.

Le système PROLOGUE étant en général distribué sur disquettes, le chargement initial du système, avant configuration et installation, se fera à partir de la disquette, appelée disquette système.

Après chargement du fichier système, une bannière de la forme suivante apparaît sur l'écran 0 :

BULL MICRAL Système d'exploitation Version 2.32 M.I. SRS1919 Taille Mémoire : 256 Ko			
PROLOGUE	Version 2.3c	ENVIRONNEMENT	Version 2.4.f
Copyright (c) 1986 BULL S.A. -> IS.SYSCONF-X- ->		Copyright (c) 1986 BULL S.A.	

La bannière environnement identifie le matériel utilisé et ses principales caractéristiques. Elle varie d'une machine à l'autre et n'est donnée ici qu'à titre indicatif.

La bannière système identifie le système PROLOGUE utilisé, version courante et no de série. Cette bannière est la même sur tous les micro-ordinateurs.

Après affichage de la bannière par le noyau système, la mise en place des modules extension est réalisée par l'utilitaire `SYSCONF` exécuté automatiquement sur le poste 0. Si l'exécution de `SYSCONF` se déroule correctement, les messages suivants sont présents sur l'écran, à la suite de la bannière système :

—> `IS.SYSCONF` —

—> _

“—>” est le symbole d'appel de PROLOGUE, indiquant qu'il est en attente d'une commande opérateur.

“IS.” spécifie que l'utilitaire `SYSCONF` et le fichier source `SYSCONF—S`, seront recherchés sur l'unité implicite système définie par la configuration préalable.

“—” en fin de ligne, indique la bonne exécution de `SYSCONF`. Tous les modules extension spécifiés dans `SYSCONF—S` ont été installés en mémoire et déclarés auprès du système.

“—> _” représente le signal d'appel, suivi du curseur.

En cas d'une erreur de configuration détectée par `SYSCONF` (module extension absent du disque système, déclaration incorrecte, ...), un message d'erreur est affiché et le processus de configuration est abandonné. Le résultat produit peut, dans ce cas, être aléatoire et il est préférable de remédier aux causes de l'erreur, avant toute exploitation du système.

REMARQUES :

1. Sur un système multiposte, la bannière système est visualisée et l'utilitaire `SYSCONF` est exécuté sur le poste 0. Les postes supplémentaires sont initialisés lorsque le traitement de `SYSCONF` est terminé. L'apparition, sur chacun des écrans, du signal d'appel “—> _”, suivi du curseur, indique la bonne initialisation du poste.
2. Si des commandes initiales ont été définies et configurées pour l'un ou l'autre poste, l'exécution de ces commandes est enchaînée après l'exécution de `SYSCONF`, ceci indépendamment sur chacun des postes utilisateur.
L'usage d'une commande initiale permet d'activer une application automatiquement, sans que l'utilisateur n'ait de commande à fournir au clavier.

1.2.3. Mise à jour de la date et de l'heure

Le système PROLOGUE entretient la date et l'heure courante, à la seconde près. Cette information est disponible pour les programmes d'application qui en ont besoin.

Lorsque le système est chargé en mémoire, la date et l'heure système sont systématiquement initialisées à une valeur par défaut.

Si ces informations présentent une importance pour votre exploitation, il vous est possible de mettre à jour la date et l'heure courante du système, au moyen de l'utilitaire DATE.

Cet utilitaire peut être également installé en tant que commande initiale, afin de garantir la mise à jour systématique de ces informations.

Pour la mise en œuvre de cet utilitaire, reportez-vous au chapitre 4.

1.2.4. Les Unités Implicites

1.2.4.1. Définition

Les unités implicites sont de trois types :

Implicite PROGRAMME pour le premier fichier d'une commande (IP)

Implicite FICHIER pour les autres fichiers (IF)

Implicite IMPRIMANTE pour la première imprimante (IM)

Ces trois implicites sont définis pour chaque poste.

Une unité implicite est une unité de support magnétique ou une imprimante définie lors de la configuration du système.

Elle est implicite, car selon la nature de la commande lancée, le système ira, si vous ne spécifiez pas la mnémonique du support, chercher par défaut l'une de ces unités.

Exemple :

—> CP <rc>

charge la commande "CP" à partir de l'implicite PROGRAMME.

1.2.4.2. Modification des unités implicites

La configuration de votre système définit des unités implicites PROGRAMME, FICHIER et IMPRIMANTE, pour chacun des postes utilisateurs (voir paragraphe 1.2.1.2.).

Si vous souhaitez temporairement modifier ces unités implicites, vous pouvez le faire à l'aide de l'utilitaire VPPI. La modification a lieu directement en mémoire et n'est pas conservée si le système est rechargé.

Pour la mise en œuvre de cet utilitaire, reportez-vous au chapitre 4.

1.2.5. Prémarquage des supports disque

Le prémarquage d'un support magnétique est l'écriture initiale sur toute la surface de ce support d'informations spécifiant l'organisation du support en pistes et secteurs et permettant un usage ultérieur du support pour les opérations courantes de lecture ou d'écriture. Cette opération est donc à effectuer la première fois que vous utilisez un disque neuf.

Le prémarquage d'un disque fixe peut durer une dizaine de minutes. Il est effectué en général une fois pour le disque.

Le prémarquage d'une disquette sera à faire pour chaque disquette neuve que vous utiliserez ou pour chaque disquette recyclée.

ATTENTION :

Le prémarquage DETRUIT totalement le contenu éventuel du support ! C'est donc une opération dangereuse que vous ne devez lancer qu'après réflexion et en ayant la certitude que le support concerné est neuf ou ne contient aucune information d'une quelconque valeur.

Le prémarquage d'un support s'effectue, soit par la commande
"→ CP,CV,...,PM",
création de volume avec prémarquage préalable (option PM), soit par la
commande
"→ CPS,...,PM",
duplication physique de support avec prémarquage préalable du support
destination (option PM).

Les commandes CP et CPS sont décrites au chapitre 4.

1.2.6. Sauvegarde de vos fichiers

La sauvegarde des fichiers que vous créez est une opération TRES importante, surtout lorsqu'ils sont volumineux du fait des informations qu'ils contiennent.

C'est donc à vous de décider à quel moment il est nécessaire d'en faire une sauvegarde.

Les causes possibles d'une destruction d'un ou plusieurs fichiers sont :

- Une panne interne à l'ordinateur pouvant provoquer des phénomènes non maîtrisable.

Exemple :

- Panne mémoire, lecteur de disquette hors service, etc ...
- Une erreur de manipulation ou de commande de votre part.

Exemple :

Utilisation d'un mauvais volume, désignation d'unité incorrecte, etc ...

La sauvegarde de vos fichiers consiste à faire une copie complète de ces fichiers sur un support disque ou disquette qui sera en général amovible et que vous pourrez par conséquent, ranger en lieu sûr.

La sauvegarde doit être effectuée régulièrement afin que le décalage entre les informations sauvegardées et les informations en exploitation reste raisonnable.

Une opération de sauvegarde peut être réalisée de différentes façons :

- Sauvegarde logique d'un volume (utilitaire CP)
- Sauvegarde logique d'un fichier ou d'un groupe de fichiers (utilitaire CP)
- Duplication physique d'un support disque ou disquette (utilitaire CPS)
- Sauvegarde d'un ensemble de fichiers sur une série de volumes destination (utilitaire SV)

Les utilitaires CP, CPS et SV sont décrits au chapitre 4.

D'autres programmes de sauvegarde, spécialisés, peuvent s'ajouter à cette liste.

(Exemple : Sauvegarde de Fichiers séquentiel-indexés).

REMARQUE IMPORTANTE :

Une sauvegarde est une opération préventive qui doit être exécutée régulièrement.

Avant une sauvegarde, assurez-vous du bon fonctionnement général de votre installation et de la validité des fichiers que vous souhaitez sauvegarder.

Dans le doute et pour ne pas détruire une sauvegarde antérieure valide, il est préférable de faire la sauvegarde de vos fichiers sur des supports différents.

Pour éviter d'avoir à vous servir de vos sauvegardes, voici quelques conseils qui pourront vous aider :

- Maintenez toujours votre ordinateur dans un état de propreté parfait.
- De manière générale, suivez les consignes d'utilisation du matériel.
- Prenez soin des disquettes que vous utilisez.
- Vérifiez régulièrement le bon fonctionnement de vos lecteurs de disques et disquettes, au moyen du programme utilitaire STATUS décrit dans le chapitre 4.
- Dans toutes les commandes que vous fournissez au système et plus particulièrement dans les commandes destructrices (prémarquage, ...), soyez attentif aux paramètres de la commande et prenez un peu de temps avant de valider la commande.

Quelques exemples de commandes de sauvegarde :

1. Sauvegarde logique d'un volume

—> CP,DV,fl0,fl1<rc>

Tous les fichiers de l'unité "FL0" sont dupliqués sur l'unité "FL1". Si un fichier de nom identique existe préalablement, son contenu est supprimé et remplacé par celui du fichier origine.

2. Sauvegarde physique d'un volume

—> CPS,fl0,fl1<rc>

Il y a duplication complète des informations du support monté en "FL0" sur le support monté en "FL1", piste à piste. Cette opération ne peut avoir lieu que pour des supports de nature identique.

3. Sauvegarde des fichiers d'un volume disque

—> SV,md0,*—*,fl0<rc>

Cette commande permet la sauvegarde de l'ensemble des fichiers du volume situé sur le disque fixe "MD0", sur une série de disquettes sauvegarde située sur "FL0". L'intérêt de cette commande est de permettre la sauvegarde de fichiers dont la taille excède la taille du support de destination.

Les volumes sauvegarde obtenus ne peuvent être relus que par l'utilitaire de restauration "RT" et ne doivent pas être utilisés dans une exploitation courante.

1.2.7. Arrêt du système

Pour arrêter votre micro-ordinateur, assurez-vous que le signal d'appel de Prologue "—>" est bien affiché sur votre écran, suivi du curseur, indiquant que le système est en attente d'une commande opérateur.

A ce stade, vous êtes assuré que les travaux de votre console sont achevés et que les fichiers sont dans un état cohérent.

Attention cependant, si vous êtes dans une configuration multiposte ou réseau, vous n'êtes pas le seul concerné par l'arrêt du système ! De même si vous faites usage du spouleur imprimante ou de toute autre tâche de fond, le retour au signal d'appel n'implique pas forcément la fin d'activité du système.

Si le SPOUL est actif, par exemple, vous devrez le désactiver avec la commande

—> SPOOL, DS <CR>

Pour mettre le système hors tension, désactivez d'abord les unités périphériques, ouvrez les portes des lecteurs de disquettes ou de cartouches de disque et retirez les volumes qui s'y trouvent. Vous pouvez alors mettre hors tension l'ensemble du système.

2.	L'ORGANISATION DES FICHIERS	2-1
2.1.	Généralités	2-1
2.1.1.	Introduction	2-1
2.1.2.	Les périphériques disques	2-2
2.1.2.1.	Organisation des supports disque	2-2
	Organisation physique	2-2
	Organisation logique	2-3
2.1.2.2.	La gestion des fichiers disques	2-7
2.1.3.	Les autres périphériques	2-9
2.1.3.1.	La périphérie caractère	2-9
	Clavier	2-9
	Ecran	2-9
	Imprimantes	2-10
2.1.3.2.	Les périphériques spéciaux	2-10
2.2.	Désignation et organisation des fichiers	2-13
2.2.1.	Désignation des unités de disque	2-13
2.2.1.1.	Identification des unités de disques	2-13
2.2.1.2.	Table des mnémoniques	2-16
2.2.2.	Désignation des fichiers	2-17
2.2.2.1.	Syntaxe générale	2-17
2.2.2.2.	Types de fichiers et suffixes conseillés	2-21
2.2.2.3.	Les caractères de substitution	2-24
2.2.2.4.	Gestion de la date de création/mise à jour	2-28
2.2.2.5.	Attribut de création d'un fichier	2-29
2.2.2.6.	Compatibilité des volumes PROLOGUE	2-29
2.2.3.	L'accès aux fichiers	2-31
2.2.3.1.	L'accès relatif	2-33
2.2.3.2.	L'accès séquentiel	2-33
2.2.3.3.	L'accès séquentiel indexé	2-33
2.2.3.4.	L'accès multi-critères	2-40
2.2.3.5.	Les bases de données	2-47

2. L'ORGANISATION DES FICHIERS

2.1. GENERALITES

2.1.1. Introduction

Un fichier est une collection d'informations reliées entre elles par des liens de nature logique et rangées sur une unité périphérique de disque. Tous les programmes et données qui interviennent dans votre exploitation se présentent sous forme de fichiers dont les caractéristiques et l'organisation interne varient en fonction de leur contenu et de leur exploitation. Une des tâches les plus importantes de PROLOGUE est d'assurer la gestion convenable de vos fichiers. Ainsi, vous n'avez pas à vous préoccuper des détails physiques concernant les opérations élémentaires de lecture ou d'écriture sur les supports disque. Vous êtes cependant responsable des ordres concernant l'exploitation de vos fichiers que vous donnez à PROLOGUE. Si vous souhaitez par exemple supprimer un fichier, il est préférable d'en donner la désignation exacte afin qu'un autre fichier ne soit pas supprimé par erreur.

La plupart des commandes relatives à la gestion des fichiers apparaît, soit dans les programmes d'application mis en œuvre, soit dans les utilitaires de PROLOGUE décrits dans le chapitre 4.

Les utilitaires CP, /, CPS, SV, RT, ... ont pour fonction principale de vous aider à gérer correctement vos fichiers.

Pour mieux les utiliser, il est important que vous soyez familiarisé avec les notions développées dans la suite de ce chapitre.

2.1.2. Les périphériques disques

2.1.2.1. Organisation des supports disque

— Organisation physique

Une unité de disque se compose d'un mécanisme de lecture/écriture, connecté à l'ordinateur, et du disque lui-même qui peut être :

- Soit un disque fixe constitué de plusieurs plateaux et intégré à l'unité de manière fixe.
- Soit une disquette ou disque souple qui peut être insérée ou retirée de l'unité suivant les besoins.
- Soit une cartouche rigide qui est également amovible.

L'intérêt de la disquette est d'être un support peu onéreux et permettant le stockage et la diffusion du logiciel. Mais sa capacité est limitée. Les capacités les plus courantes vont de 300 000 caractères à un million de caractères.

L'intérêt du disque fixe est d'offrir une capacité de stockage importante de une ou plusieurs dizaines de millions de caractères. Son défaut est d'être fixe, ce qui impose des procédures de sauvegarde fastidieuses.

La cartouche rigide est un intermédiaire très confortable, offrant des capacités élevées et simplifiant les procédures de sauvegarde. C'est cependant un type d'unité encore peu répandu en micro-informatique.

Une unité de disque, quelle que soit sa nature, se caractérise par la présence d'une surface magnétique sur un ou plusieurs plateaux en rotation permanente, sur lesquels sont enregistrées les informations. L'accès aux informations est assuré par des têtes de lecture/écriture, une par face, qui peuvent se déplacer radialement. Chaque plateau est divisé en pistes concentriques et chaque piste en un certain nombre de secteurs de capacité égale, destinés à contenir des informations que l'utilisateur pourra consulter ou modifier à son gré. Le secteur est donc l'unité PHYSIQUE d'échange entre le disque et l'ordinateur. La taille d'un secteur est le plus couramment de 256 à 512 caractères. Le nombre de faces, le nombre de pistes par face, le nombre de secteurs par piste et la taille des secteurs définissent les caractéristiques de l'unité périphérique et peuvent varier d'un type de périphérique à un autre et d'un ordinateur à l'autre.

Par exemple, les caractéristiques suivantes sont assez courantes :

	Nb de faces	NB de pistes/ face	Nb de secteurs/ piste
Disquette	2	75	15
Disque fixe	4	153	32

Les valeurs ci-dessus ne sont données qu'à titre indicatif, car il existe beaucoup d'autres formats de disques.

— Organisation logique

La structure des disques décrite ci-dessus représente l'organisation PHYSIQUE du disque. Cette organisation est transparente lorsque vous travaillez avec PROLOGUE. En effet, PROLOGUE prend complètement en charge l'implantation des fichiers de vos applications sur les supports disque. C'est ce qu'on appelle l'organisation LOGIQUE du support disque. Elle est propre à chaque système d'exploitation utilisé. Pour vous permettre de mieux comprendre l'action des commandes que vous adressez au système, nous allons décrire l'organisation logique des unités de disque, assurée par PROLOGUE.

L'organisation logique est constituée des éléments suivants :

VOLUME :

Le système PROLOGUE considère chaque unité de disque comme un volume unique défini par son nom et ses caractéristiques de stockage. Le volume est associé au support disque sur lequel il est installé. Si le disque est un disque fixe, il y a identité complète entre le volume PROLOGUE et l'unité périphérique elle-même. Si le disque est une disquette ou une cartouche, le volume varie suivant le support introduit et est donc variable. Pour reconnaître les volumes gérés par PROLOGUE, le système positionne, sur chaque volume, un indicatif de structure dans le troisième octet du quatrième secteur. Chaque valeur de l'indicatif représente sa structure précise et permet au système de s'y adapter (voir paragraphe 2.2.2.6).

PISTE/SECTEUR :

Pour permettre d'adresser directement un secteur précis d'un disque, ce qui est parfois nécessaire dans certaines opérations, PROLOGUE définit une décomposition logique du disque en pistes et secteurs qui ne sera pas nécessairement identique à la structure physique du disque.

Le secteur logique, défini par PROLOGUE, a une taille fixe de 256 caractères, indépendante de la taille effective des secteurs physiques. Il correspond à l'unité logique d'échange entre PROLOGUE et les unités de disque.

La piste logique définie par PROLOGUE représente le plus souvent une piste physique ou un cylindre (un cylindre = toutes les pistes correspondent à une même position des têtes de lecture/écriture). Cette décomposition est faite en fonction de critères d'efficacité. Toutes les commandes de base de PROLOGUE faisant référence à une notion de piste ou secteur adressent les pistes et secteurs logiques tels que définis ci-dessus. PROLOGUE numérote les pistes et les secteurs d'un disque, de la façon suivante :

Les pistes sont numérotées de 0 à $P-1$, P étant le nombre total de pistes du support.

Les secteurs d'une même piste sont numérotés de 0 à $S-1$, S étant le nombre de secteurs par piste du support.

Un secteur donné du disque peut être, soit désigné par le numéro de piste et le numéro de secteur dans la piste. Dans ce cas, on dit que l'adresse secteur est une adresse RELATIVE. Il peut aussi être désigné par son numéro de secteur, compté à partir du début du support. Dans ce cas, on dit que l'adresse secteur est ABSOLUE.

Exemple :

La commande suivante de l'utilitaire PATCH :

```
-> PATCH,f10,npis=10,nsec=2<rc>
```

fait référence à la piste 10 et au secteur 2 de cette piste, suivant le format défini par PROLOGUE. Si l'on suppose qu'il y a 16 secteurs par piste, l'adresse absolue du secteur est le secteur 162, c'est-à-dire $10 * 16 + 2$.

GRANULE :

Pour pouvoir stocker des données sous forme de fichiers, sur un support disque, PROLOGUE prend en charge le choix des secteurs du disque à utiliser de façon dynamique. Ce processus est complètement transparent pour l'utilisateur.

Lors de la création d'un volume, l'utilisateur a la possibilité de définir l'unité d'allocation pour ce volume. C'est ce qu'on appelle le GRANULE, c'est-à-dire le plus petit élément que l'on puisse allouer ou libérer sur le disque. La taille du granule est définie en secteurs logiques (donc de 256 caractères) et peut prendre les valeurs suivantes : 1,2,4,8,16,32,64,128, 256 secteurs. La taille attribuée par défaut est de 16 secteurs.

La commande "CV" (création de volume) de l'utilitaire "CP", décrit au chapitre 4, possède un paramètre optionnel pour spécifier la taille des granules d'un volume. L'espace occupé par un fichier est toujours un nombre entier de granules et au minimum, un granule. Le premier granule, ou granule numéro 0 du volume, commence à la piste 0 et secteur 0 logiques. Le nombre total de granules d'un volume dépend de la taille du granule et de la capacité du support disque.

REMARQUE :

La taille du granule est un paramètre attaché au volume PROLOGUE et non à l'unité périphérique qui le supporte.

Le nombre maximum de granules est de 8176.

BLOC :

Un bloc est constitué d'un ou plusieurs granules consécutifs sur le disque. Un bloc peut donc être repéré par le numéro du premier granule et le nombre de granules qui le composent (maximum 255). PROLOGUE autorise un nombre illimité de blocs pour un même fichier.

Les granules et blocs composant un fichier donné peuvent être visualisés par l'utilitaire "/" (catalogue), à l'aide de l'option "GR" (voir chapitre 4).

TABLE D'ALLOCATION :

Chaque volume débute par une table implantée en piste 0, secteur 0 du disque, d'une longueur de 1 à 4 secteurs suivant la capacité du disque, dans laquelle est indiqué l'état libre ou occupé de chaque granule du disque. A chaque granule correspond un indicateur élémentaire dans cette table (bit). Si l'indicateur est à zéro, le granule correspondant est libre. Si l'indicateur est à 1, le granule correspondant est occupé. Cette table d'allocation, ainsi que le répertoire, occupent un nombre entier de granules, en tête du volume. Ces granules sont indiqués, dans la table, comme étant occupés. PROLOGUE prend en charge la gestion complète de cette table.

REPertoire OU CATALOGUE :

Chacun des fichiers enregistrés sur un volume se compose des données propres au fichier, manipulées par les programmes et d'informations internes à PROLOGUE décrivant l'implantation, la taille et les attributs de ce fichier. Ces informations sont regroupées pour fournir le répertoire du volume mémorisé après la table d'allocation. C'est le répertoire qui permet de connaître la liste des fichiers présents sur un volume PROLOGUE. Le répertoire, encore appelé catalogue, peut être visualisé à l'aide de l'utilitaire "/".

La taille du répertoire ou catalogue est variable et définit le nombre maximum de fichiers que l'on peut créer sur le volume. Cette taille est définie lors de la création du volume (commande CP,CV,...) et ne peut être modifiée. Le nombre maximum de fichiers sur un même volume est de 895.

Les informations ci-dessus sont données pour vous permettre de mieux comprendre les mécanismes internes du fonctionnement de PROLOGUE, mais il n'est pas nécessaire de les connaître pour pouvoir utiliser vos applications.

2.1.2.2. La gestion des fichiers disques

Le Noyau de Gestion des Fichiers (N.G.F.)

Un module particulier de PROLOGUE, appelé le N.G.F. (Noyau de Gestion des Fichiers), est chargé de gérer toutes les unités de disque de votre configuration. Il prend en charge, d'une manière transparente pour l'utilisateur, l'organisation des volumes et maintient à jour les informations, en fonction des opérations effectuées sur les fichiers.

Le N.G.F. autorise les opérations suivantes sur les fichiers :

CREATION :

Cette opération consiste à introduire un nouveau fichier, dans le répertoire d'un volume. Il y a mise à jour de toutes les informations descriptives du fichier : nom, suffixe, clés d'accès, attributs, etc... Le contenu du fichier est initialement vide. Pour que le fichier puisse être créé, il est nécessaire de disposer d'une place libre dans le répertoire, sinon la création est refusée. Si un fichier de même nom et de même type existe déjà dans le répertoire du volume, la création est également refusée.

SUPPRESSION :

Cette opération est l'opération inverse de la création.

Le fichier est supprimé du répertoire du volume et toutes les données qu'il contenait sont effacées. Les granules libérés par ce fichier sont déclarés à nouveau disponibles dans la table d'allocation et seront réutilisés lors d'une prochaine allocation. Pour pouvoir être supprimé, un fichier doit être fermé.

OUVERTURE :

Cette opération consiste à déclarer le fichier auprès du système, dans le but d'accéder aux données de ce fichier.

L'ouverture doit être maintenue pendant la durée des accès réalisés. Il est possible de procéder à une ouverture **EXCLUSIVE** du fichier : dans ce cas, tout autre demande d'ouverture faite sur ce même fichier sera refusée.

L'ouverture peut également être **PARTAGEABLE** : dans ce cas, plusieurs ouvertures simultanées sont acceptées, ce qui permet un partage des données entre plusieurs utilisateurs du système.

FERMETURE :

C'est l'opération inverse de l'ouverture. La fermeture supprime la déclaration du fichier auprès du système et garantit la bonne mise à jour de toutes les données écrites dans le fichier.

LECTURE :

Cette opération permet d'accéder aux données contenues dans le fichier préalablement ouvert. Le contenu du fichier est "vu" comme une succession de secteurs logiques numérotés de 0 à $N-1$, N étant la taille du fichier en secteurs logiques.

La répartition réelle du fichier sur le disque est complètement indépendante et cachée. La lecture permet à un programme d'acquérir un nombre entier de secteurs (maximum 255 secteurs), à partir d'une adresse secteur relative au début du fichier.

ECRITURE :

Cette opération permet à un programme de mettre à jour des données dans un fichier préalablement ouvert. Si le ou les secteurs adressés correspondent à des secteurs contenus dans le fichier, il y a mise à jour du contenu de ces secteurs, sans modification de la taille totale du fichier. Si le ou les secteurs référencés font suite au dernier secteur du fichier, ils sont ajoutés au fichier dont la taille augmente d'autant.

Ces fonctions sont les fonctions élémentaires permettant de gérer les fichiers sur disque. Il n'y a pas nécessairement équivalence directe entre ces fonctions et les commandes des utilitaires de PROLOGUE. Seules les commandes de création de fichier (commandes "CP,CF, ...") et de suppression de fichier (commande "CP,SF, ...") correspondent directement aux fonctions de création et suppression du N.G.F.

Le plus souvent, c'est une combinaison de ces fonctions qui est utilisée dans les commandes de gestion de fichiers.

Le N.G.F. assure l'organisation des fichiers sur les unités de disque et permet un accès élémentaire aux données du fichier. PROLOGUE permet de gérer des structures de données plus complexes, au moyen de modules optionnels appelés méthodes d'accès, qui sont décrits au paragraphe 2.2.3.

2.1.3. Les autres périphériques

Les paragraphes précédents nous ont permis de détailler les ressources périphériques de type "disque" qui, malgré leur diversité, ont en commun d'être gérées de manière homogène par le même module N.G.F. D'autres ressources existent qui ne sont pas gérées par le N.G.F. Ce sont d'une part, les ressources de la périphérie "caractère" et les ressources "spéciales", d'autre part.

2.1.3.1. La périphérie caractère

Ce sont essentiellement les ressources clavier, écran et imprimante. Le terme "caractère" indique que les échanges entre l'ordinateur et ces ressources a lieu caractère à caractère et non pas secteur à secteur, comme c'est le cas pour les ressources disque.

De plus, par rapport à une ressource "disque", où l'accès au secteur est direct, l'accès à la périphérie caractère est nécessairement séquentiel. La notion de fichier n'existe plus pour ce type de ressource, ou plutôt elles sont "vues" comme un fichier unique, sans nom, ouvert en permanence. Ces périphériques ne sont en général pas partageables et sont donc attachés à un des postes utilisateurs du système (sauf pour l'imprimante qui peut être partageable suivant la configuration du système. Voir chapitre 7).

LE CLAVIER

Ce périphérique n'est utilisable qu'en entrée (ou lecture).

Un module particulier du système permet aux programmes exécutés de lire les caractères frappés au clavier. Le clavier peut être vu comme un fichier ouvert en permanence, dans lequel on peut lire séquentiellement une suite de caractères.

L'ECRAN

Ce périphérique n'est utilisable qu'en sortie (ou écriture). Un module particulier du système permet aux programmes exécutés d'écrire des caractères sur l'écran. L'écran peut être vu comme un fichier ouvert en permanence dans lequel on peut écrire en ajout et séquentiellement une suite de caractères.

L'ensemble clavier/écran est individuel pour chacun des utilisateurs du système et prend le nom de CONSOLE utilisateur. La console permet de "piloter" l'exploitation de l'ordinateur et de "dialoguer" avec PROLOGUE.

L'IMPRIMANTE

Ce périphérique n'est utilisable qu'en sortie et permet d'obtenir une trace écrite des données. Suivant la configuration du système, ce périphérique est, soit vu comme un périphérique de type "caractère" propre à un poste utilisateur, soit vu comme une ressource spéciale partageable (voir paragraphe suivant).

Les ressources caractères décrites ci-dessus sont les plus courantes sur un micro-ordinateur, mais il en existe d'autres moins fréquentes : souris, crayon optique, entrée/sortie vocale, etc...).

2.1.3.2. Les périphériques spéciaux

On désigne par le terme de ressource "spéciale", les ressources périphériques, partageables ou non, qui s'adressent d'une façon similaire à un fichier, mais qui, compte tenu de leurs caractéristiques fonctionnelles, sont gérées par des modules spéciaux et non plus par le N.G.F. Par exemple, un dérouleur de bande magnétique, par le caractère séquentiel de son organisation, est géré par un module particulier qui lui est propre.

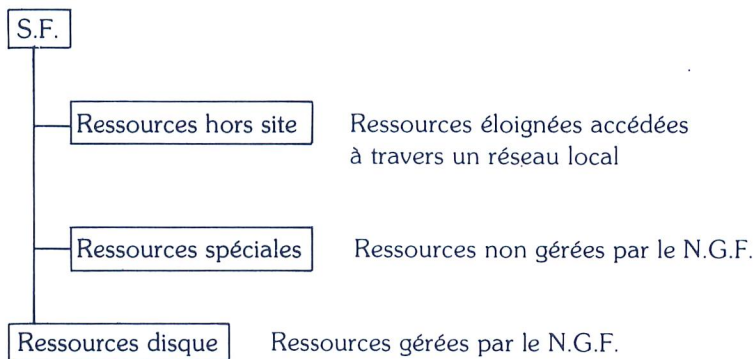
L'accès à une ressource spéciale se fait par des fonctions de type "fichier". On y retrouve donc des ordres d'ouverture, de lecture, écriture et fermeture, qui prennent des significations particulières suivant la nature du périphérique.

Prenons l'exemple d'une voie de télécommunications. L'établissement de la connexion se fera par un ordre d'ouverture, une émission de message par un ordre d'écriture, une réception de message par un ordre de lecture. La déconnexion se fera par un ordre de fermeture.

Le même type de fonction est donc utilisé aussi bien pour l'accès à des fichiers sur disque que pour l'accès à des ressources spéciales.

Cette généralisation de la notion de fichiers permet d'assurer une interface homogène pour l'accès à l'ensemble des ressources périphériques. C'est ce qu'on appelle l'interface SF (système de fichiers).

La structure du système de fichiers peut être décomposée de la façon suivante :



Suivant la nature de la ressource désignée, la requête d'appel est aigüillée vers le module adéquat du système de fichiers.

La mise en œuvre de ressources spéciales est liée à des configurations et des usages particuliers et il est nécessaire de se reporter aux documentations spécifiques attachées à ces ressources.

2.2. DESIGNATION ET ORGANISATION DES FICHIERS

2.2.1. Désignation des unités de disque

2.2.1.1. Identification des unités de disques

Il est nécessaire de pouvoir identifier et désigner les différentes unités de disque de votre configuration, soit dans les commandes des utilitaires de PROLOGUE, soit dans les programmes d'application. La désignation de l'unité est nécessaire, soit pour pouvoir accéder à un fichier particulier ou un groupe de fichiers sur un volume PROLOGUE précis, soit pour effectuer une opération globale sur le volume ou sur la ressource périphérique.

Remarquez bien que la désignation de l'unité identifie physiquement le périphérique et non pas le volume PROLOGUE lui-même. Dans le cas d'une unité de disque fixe, il y a équivalence complète, mais dans le cas d'un support amovible, le volume adressé est celui installé sur l'unité, lors des accès. Nous reviendrons sur ce point, un peu plus loin.

La désignation d'une unité périphérique se compose d'une mnémonique de périphérique, suivi d'un numéro d'unité. La mnémonique est composée de deux lettres alphabétiques identifiant la nature du périphérique.

Exemple :

“FL” pour unité de disque souple ou (FL)oppy
“MD” pour l'unité de disque fixe ou (M)ini-(D)isque
etc...

Les caractères de la mnémonique peuvent être frappés en position MAJuscule ou minuscule. Il y a équivalence entre les deux.

Le numéro d'unité est composé d'un chiffre entre 0 et 7 inclus, permettant d'identifier plusieurs unités de même type, ayant donc la même mnémonique. Pour une configuration composée par exemple de deux unités de disques souples, la première des unités sera désignée par “FL0”, l'autre unité par “FL1”. Pour PROLOGUE, ce sont deux unités périphériques totalement séparées. Cependant, l'emploi d'une mnémonique commune permet d'indiquer immédiatement la nature identique de ces deux unités.

Les mnémoniques attribuées aux différentes unités de votre ordinateur dépendent de la configuration et de l'adaptation particulière de PROLOGUE à ces matériels. On peut donc imaginer les mnémoniques les plus diverses, pourvu qu'elles respectent la syntaxe décrite ci-dessus. Cependant, l'usage veut que l'on retrouve le plus souvent les désignations suivantes :

- Configuration : deux unités de disques souples
Les unités définies par PROLOGUE seront :
“FL0” pour la première unité,
“FL1” pour la deuxième unité.
- Configuration : une unité de disque fixe, une unité de disque souple
Les unités définies par PROLOGUE seront :
“MD0” pour l'unité de disque fixe,
“FL0” pour l'unité de disque souple.

La désignation d'une unité dans les commandes d'appel des utilitaires de PROLOGUE se fait :

- Soit SEULE, lorsque l'opération concerne un volume ou un support périphérique dans son ensemble.

Exemple :

—> CP,CV,fl0,nomvol<cr>

L'opération de création de volume est globale. La commande précise sur quelle unité est situé le support disque sur lequel il faut créer un volume PROLOGUE.

- Soit ASSOCIÉE à une désignation de fichier ou d'un groupe de fichiers, lorsque l'opération est spécifique à ce ou ces fichiers. Dans ce cas, la désignation de l'unité est séparée du nom de fichier par un point, “.”.

Exemple :

—> CP,DF,fl0.fic1,md0.fic2<rc>

Cette opération consiste à dupliquer le fichier "fic1" situé sur l'unité "FL0" vers le fichier "fic2" situé sur l'unité "MD0". Remarquez que lorsque vous localisez le fichier "fic1" ou "fic2" dans cette commande, vous n'indiquez pas le nom du volume dans lequel le fichier est présent, mais plutôt l'unité supportant ce volume, lors de l'opération. Le nom d'un volume est une information utile pour vous permettre de mieux identifier son contenu, mais elle n'est pas directement utilisée pour leur localisation, dans les commandes d'appel.

Dans la description des commandes d'appel du chapitre 4, le terme "unité1", "unité2", "unité3", etc... est utilisé pour indiquer que vous devez fournir à cet endroit une désignation d'unité, par exemple "FL0".

Note :

Si le nom du support est suivi d'un point et d'un nom de fichier, le chiffre zéro est utilisé par défaut

Exemple :

FL.COMPTA
est équivalent à
FL0.COMPTA

2.2.1.2. Table des mnémoniques

Mnémonique	Signification
FL0	Disquette 5" 1/4 : unité 0
FL1	Disquette 5" 1/4 : unité 1
FS0	Disquette 8" : unité 0
FS1	Disquette 8" : unité 1
MD0	1 ^{er} Mini-disque intégré
MD1	2 ^{eme} Mini-disque intégré
MD2	3 ^{eme} Mini-disque intégré
.	.
.	.
.	.
.	.
MDn	n ^{eme} mini-disque intégré
DS0	Disque externe fixe
DS1	Disque externe amovible
DS2	2 ^{eme} disque externe fixe
DS3	2 ^{eme} disque externe amovible
GS0	Disque externe 60 Mo : unité 0
GS1	Disque externe 60 Mo : unité 1
VD0	Disque externe 57 Mo : unité 0
VD1	Disque externe 57 Mo : unité 1
WD0	Disque externe 10 Mo

2.2.2. Désignation des fichiers

2.2.2.1. Syntaxe générale

Sous PROLOGUE, la désignation d'un fichier se fait sous la forme générale suivante :

[unité.]nomfic[—suf][:illee]

avec :

UNITE : Désignation de l'unité de disque supportant le volume contenant le fichier "nomfic—suf". La syntaxe de "unité" suit les règles données au paragraphe 2.2.1. Ce champ est facultatif et doit être séparé du nom de fichier par un point (caractère "."). Dans le cas où la désignation de l'unité est omise, une unité de disque est choisie par défaut, suivant le rôle joué par le fichier dans la commande d'appel. Si le fichier désigne le nom du programme à exécuter et est de type —X, c'est l'unité implicite programme du poste qui est choisie par défaut. Si le fichier désigne un fichier de données en paramètre de la commande, c'est l'unité implicite fichier du poste qui est choisie par défaut.

Exemple :

—> CP,DF,ficdat—s,CO<rc>

Cette commande affiche à l'écran le contenu du fichier "ficdat—s". "CP" désigne le programme utilitaire CP à exécuter. Par conséquent, le fichier correspondant sera recherché sur l'unité implicite programme de votre poste.

"ficdat—s" est un fichier en paramètre de la commande.

Il sera donc recherché sur l'unité implicite fichier de votre poste. Ce mécanisme d'unités implicites est très commode car il permet le plus souvent d'éviter d'avoir à donner la désignation des unités dans les commandes d'appel.

La façon de configurer ses unités implicites de poste est décrite au chapitre 1. Si vous souhaitez connaître vos implicites de poste, vous pouvez, à tout moment, appuyer sur la touche retour chariot (<rc>), immédiatement à la suite du signal d'appel de PROLOGUE :

—> <rc>

Vos implicites de poste seront alors affichés. Par exemple :

—> Prologue Version 2.3 Poste 0 Implicites FL0.FL1.IM0

Reportez-vous au paragraphe 3.3.3, pour plus de détails.

NOMFIC : Représente le nom du fichier. "nomfic" se compose de 1 à 8 caractères, sans espace. Les caractères autorisés sont les suivants :

Caractères alphabétiques: de "A" à "Z" et de "a" à "z"

Caractères numériques: de "0" à "9"

Caractères spéciaux: "\$", "/", "+", "@", "#", "%", "_"

Le caractère blanc " " est interdit, car il fait office de séparateur. Les "*" et "?" peuvent être utilisés dans la désignation d'un fichier mais jouent un rôle particulier, décrit ci-dessous. Notez que le nom d'un fichier peut être indifféremment frappé en minuscules ou en majuscules, les minuscules étant assimilées aux majuscules dans la codification du nom.

D'autre part, les caractères de type national "é", "à", ... sont interdits.

"nomfic" est un champ obligatoire dans la désignation d'un fichier et ce peut être le seul.

Exemple :

—> EDV.FPAIE<rc>

Cette commande désigne implicitement le fichier "fL1.FPAIE-s", si l'unité implicite fichier est "FL1", mais seul le champ NOMFIC a été utilisé comme paramètre d'appel de EDV.

IMPORTANT

Lorsque vous devez désigner une unité seule dans une commande, PROLOGUE ne vous impose pas de faire suivre cette désignation par le caractère ".", ce qui allège la désignation. En revanche, il est possible de créer une ambiguïté entre un nom de fichier et un nom d'unité. Si par exemple "FL" est une ressource connue du système, le nom de fichier "FLA" ne sera pas admis par PROLOGUE, alors que le nom "FLAP" le sera.

À l'inverse, si la ressource "MD" n'est pas déclarée auprès du système, le nom de fichier "MD0" sera parfaitement admis.

Il est donc recommandé d'utiliser des noms de fichiers qui soient nettement différents d'une désignation possible d'unité afin d'éviter tout risque d'ambiguïté.

SUF : C'est un champ facultatif appelé SUFFIXE, qui caractérise la nature du contenu du fichier. Le suffixe se compose de 1 à 3 caractères, sans espace. Les caractères autorisés pour le suffixe sont identiques à ceux autorisés pour le nom du fichier. Le suffixe est séparé du nom de fichier par un tiret (caractère "-"). Lorsque le champ "suf" est omis dans la désignation d'un fichier, il prend une valeur par défaut, suivant la nature de la commande et le rôle joué par le fichier dans cette commande. Cette valeur par défaut peut être le caractère "blanc" et le fichier est alors sans suffixe ou avec le suffixe "blanc".

Le paragraphe suivant décrit les suffixes associés aux types standards de fichiers reconnus par PROLOGUE.

Remarques :

C'est l'ensemble "NOMFIC-SUF" qui identifie exactement le fichier sur un volume PROLOGUE. Sur un même volume, il ne peut y avoir deux fichiers ayant le même nom Et le même suffixe.

Par contre, le même nom et le même suffixe peuvent être employés sur deux volumes différents.

NOMFIC et SUF sont enregistrés dans le catalogue du volume, lors de la création du fichier.

Le champ "unité" permet de localiser le fichier, mais ne l'identifie en aucune manière.

D'autre part, il est possible d'utiliser le caractère "." comme séparateur entre un nom de fichier et son suffixe, à la place du tiret. Cette notation est admise par PROLOGUE pour des raisons de compatibilité avec la syntaxe des décors CP/M-86 et MS-DOS (voir chapitre 6).

La désignation "FL0.fic001.src" est donc parfaitement valable. Cependant, cette notation n'est pas la notation naturelle de PROLOGUE et ne doit être utilisée que lorsque vous êtes amené à utiliser de manière significative les décors.

Le suffixe est un moyen commode d'identifier le TYPE d'un fichier. Formellement, le suffixe est une notion syntaxique, élément de la désignation d'un fichier, alors que le type est une notion sémantique attachée au contenu du fichier.

Par exemple, au type binaire sont attachés différents suffixes : "X", "EXE", etc...

Mais en général, les deux notions coïncident et il est fréquent que par abus de langage, on parle du fichier de type "X", alors qu'il faudrait parler du fichier de type binaire ayant le suffixe "X".

LLEE : Ce champ facultatif désigne deux clés de protection permettant de protéger l'accès au fichier. Chaque clé est composée de deux caractères, blanc inclus. La première clé, "LL", est la clé de protection en lecture. Elle doit être fournie correctement dans la commande d'appel, pour que des accès en lecture soient possibles. La deuxième clé "EE", est la clé de protection en écriture et suppression. Elle doit être fournie correctement dans la désignation du fichier pour que des accès en écriture ou que la suppression du fichier soient possibles. Ces deux clés peuvent être spécifiées indépendamment. Les caractères autorisés pour une clé de protection sont les mêmes que ceux autorisés pour un nom de fichier. Cependant, un blanc peut être utilisé en premières positions. Par exemple, la clé "X" est valide.

Lorsque les clés d'accès sont omises, leur valeur par défaut est le blanc (deux caractères blancs). Lorsqu'un fichier est créé sans clés de protection, il est donc créé avec des clés à blanc et toutes les désignations ultérieures du fichier peuvent être faites sans désignation de clés.

Remarques :

Les clés d'accès ne participent pas à l'identification d'un fichier. De plus, pour leur donner un caractère confidentiel, vous remarquerez que les clés d'accès d'un fichier ne sont jamais affichées à l'écran par la commande de catalogue ("/"), ou toute autre commande du système.

- “EXE” Programme binaire exécutable par le décor MS.DOS.
ou “COM”
- “CMD” Programme binaire exécutable par le décor CP/M-86.
- “T” Programme traduit BAL exécutable par l'interpréteur BAL (EX-X).
- “ASG” Fichier de type ASCII, éditable, contenant une série de commandes exécutables par le programme utilitaire ASG.
- “S” fichier de type ASCII, éditable, contenant généralement le texte source d'un programme, par exemple un programme BAL.
- “I” Fichier INDEX d'un fichier séquentiel indexé.
- “D” Fichier DONNEES d'un fichier séquentiel indexé.
L'ensemble des fichiers de suffixe I+D constitue un fichier séquentiel indexé unique, géré par la méthode d'accès correspondante (se reporter au paragraphe 2.2.3.).
- “U” Fichier de sauvegarde d'un fichier de type séquentiel indexé.
- “V” Fichier VALEUR d'un fichier multicritère.
- “R” Fichier REFERENCES d'un fichier multicritère.
L'ensemble des fichiers de suffixe I,D,V,R constituent un fichier multicritère unique, géré par la méthode d'accès correspondante (se reporter au paragraphe 2.2.3.).
- “B” Fichier DICTIONNAIRE d'un fichier Base de Données.
L'ensemble des fichiers de suffixe B,V,R,D,I, constitue un fichier base de données unique, géré par la méthode d'accès correspondante (se reporter au manuel de référence de la base de données).
- “A” Fichier Ancien.
Version précédente d'un fichier texte “Sources” après mise à jour par l'Editeur de Textes Vidéo, EDV.

Il existe d'autres suffixes de fichier ayant une valeur implicite. Certains suffixes sont spécifiques à une application ou un programme précis et ne sont donc pas détaillés ici. Reportez-vous à la documentation attachée au produit, pour en connaître la signification précise.

La liste des suffixes ci-dessus fait apparaître trois types très courants de fichiers, que nous allons détailler :

Fichier TRADUIT :

Contient un programme exécutable par l'exécuteur BAL.

Il contient des instructions que l'exécuteur BAL est capable d'interpréter et d'exécuter. C'est le cas des fichiers de suffixe "T".

Fichier ASCII ou TEXTE :

C'est un fichier composé d'une suite de caractères codés suivant la norme PROLOGUE ASCii étendu) et pouvant être affichés à l'écran ou imprimés. La création ou la mise à jour de ce type de fichier se fait à l'aide d'un programme appelé Editeur de textes (Exemple : EDV). Un fichier ASCII contient souvent le texte d'un programme, c'est-à-dire la liste des instructions du programme. Dans ce cas, on parle de fichier SOURCE. C'est le cas par exemple, d'un fichier de suffixe "S".

L'usage des suffixes implicites est très intéressant à deux titres :

1. Cela simplifie les commandes utilisateurs désignant des fichiers. Vous n'avez pas à spécifier le suffixe du fichier, cela est fait automatiquement par le programme.
2. Le type du fichier est immédiatement visible. Lors de l'affichage du catalogue d'un volume, le type des fichiers présents est immédiatement déterminé par le suffixe de ces fichiers.

PROLOGUE vous permet cependant d'utiliser un suffixe quelconque pour un fichier, quel que soit son type. Rien ne vous empêche par exemple, de désigner un fichier source par "FCOMPT-BIN" et d'éditer ce fichier à l'aide de l'éditeur EDV ! Mais par contre, vous perdez alors le bénéfice du suffixe implicite. Votre commande d'appel sera nécessairement :

"-> EDV,FCOMPT-BIN<rc>" et non pas "-> EDV,FCOMPT<rc>

Fichier BINAIRE :

Contient un programme exécutable par PROLOGUE.

Le terme "binaire" indique que le fichier contient une série d'instructions "machine", c'est-à-dire directement exécutables par le micro-processeur de la machine. C'est le cas par exemple, des fichiers de suffixe "X" ou "EXE".

De plus, vous augmentez les risques de confusion sur la nature du fichier. Il est donc recommandé, autant que possible, de respecter les suffixes implicites de fichiers définis par PROLOGUE.

2.2.2.3. Les caractères de substitution

Nous venons de voir les principes de désignation d'un fichier. Certaines commandes de PROLOGUE permettent de désigner, non pas UN fichier, mais un groupe particulier de fichiers sur lequel s'applique la commande. Par exemple, si vous souhaitez faire une copie de l'unité de disque fixe "MD0" vers l'unité de disquette "FL0", de tous les fichiers SOURCE BAL pour en faire la sauvegarde, il est très pratique de pouvoir désigner en une seule commande, tous les fichiers dont le type est "S" plutôt que de dupliquer un à un les fichiers de type "S", avec le risque d'en oublier un ! Pour permettre la désignation d'un groupe de fichiers, deux caractères spéciaux sont utilisés, "*" (étoile) et "?" (point d'interrogation). Ils sont donc interdits en tant que caractères propres d'un nom de fichier. Leur utilisation se fait en remplacement de caractères composant le nom ou le suffixe d'un fichier, à l'exclusion de l'unité et des clés de protection. Tous les fichiers composant un groupe désigné de cette façon sont nécessairement situés sur le même volume PROLOGUE, donc sur la même unité de disque.

Nous allons voir leur usage respectif :

Le caractère étoile "*"

Ce caractère permet le remplacement d'un ou plusieurs caractères composant le nom ou le suffixe d'un fichier. Lorsqu'il est utilisé dans un nom de fichier, il doit être cadré à droite ou à gauche du nom et complété éventuellement par un maximum de sept caractères. Il permet de désigner tous les fichiers dont le nom coïncide avec la partie des caractères donnés explicitement et dont le reste est quelconque, éventuellement vide. Il peut être utilisé seul et désigne dans ce cas, un nom quelconque de fichier. Il ne peut être utilisé qu'une fois dans le nom de fichier. Donnons quelques exemples :

"CG*" désigne tous les fichiers dont le nom commence par "CG".
Par exemple : "CGROS" ou "CG".

“*A” désigne tous les fichiers dont le nom se termine par “A”.
Par exemple : “XYZA”.

“*” désigne tous les fichiers de nom quelconque et dont le suffixe doit être blanc.

Le caractère étoile est utilisé dans le suffixe du fichier, de la même façon qu’il l’est dans le nom.

Exemple :

“MONPROG—*” désigne tous les fichiers dont le nom est “MONPROG”,
mais dont le suffixe est quelconque.
Par exemple : “MONPROG—TYP”.

“FICOR—X*” désigne tous les fichiers dont le nom est “FICOR” et
dont le suffixe commence par un “X”.
Par exemple : “FICOR—X” ou “FICOR—X\$\$”.

L’usage du caractère “*” peut être combiné dans le nom et le suffixe, ce qui permet une sélection précise d’un groupe de fichiers. Dans ce cas, les fichiers référencés sont ceux dont le nom et le suffixe satisfont simultanément aux conditions de sélection.

Exemple :

“*/PC—*” désigne tous les fichiers dont le nom se termine par
“/PC” et dont le suffixe est quelconque.

“QWERTY*—T” désigne tous les fichiers dont le nom commence par
“QWERTY” et dont le suffixe est “T”.

“*—*” désigne tous les fichiers d’un Volume PROLOGUE.

Le caractère point d’interrogation “?” :

Ce caractère permet de remplacer, un à un, un nombre quelconque de caractères composant le nom ou le suffixe d’un fichier. Tous les fichiers dont le nom et le suffixe ne coïncident que pour les caractères explicites donnés dans la désignation, seront sélectionnés par la commande correspondante. Un nombre quelconque de points d’interrogation peuvent être utilisés simultanément dans le nom et le suffixe, à concurrence de huit pour le nom et trois pour le suffixe. Le caractère “?”, si c’est le dernier caractère de la désignation, ou s’il n’est suivi que par d’autres caractères “?”, peut ne correspondre à aucun caractère dans le nom effectif du fichier qui sera donc plus court que la désignation de groupe.

En d'autres termes, un groupe de n caractères "?" consécutifs, situés à droite de la désignation (exemple : "abc???") peut remplacer effectivement zéro à n caractères. Par contre, ceci n'est pas vrai si le groupe de "?" est cadré à gauche.

Par comparaison avec le caractère étoile, le point d'interrogation ne remplace qu'un et un seul caractère de la désignation, mais en revanche, il peut être employé de façon multiple dans une même désignation.

Exemple :

- | | |
|----------------|--|
| "?NNN-?TT" | désigne tous les fichiers dont le nom commence par deux caractères quelconques, autres que le blanc, suivis par les caractères "NNN" et dont le suffixe se compose d'un caractère quelconque, autre que le blanc, suivi par les caractères "TT". Le nombre de caractères composant le nom est nécessairement de cinq, celui composant le suffixe est de trois. |
| "?abcdef?-xyz" | désigne tous les fichiers dont le nom commence par un caractère quelconque autorisé, mais PAS le blanc, suivi par les caractères abcdef, puis d'un caractère quelconque autorisé, éventuellement le blanc, et donc le suffixe est "xyz".
Par exemple : "Zabcdef-xyz" ou "/abcdef/-xyz". |
| "PROL????-REF" | désigne tous les fichiers dont le nom commence par PROL, suivi par zéro à quatre caractères et dont le suffixe est "REF". Cette notation est équivalente à la notation "PROL*" - "REF". |
| "????????-???" | désigne tous les fichiers d'un même volume. Cette désignation est équivalente à "*-*". |

Les caractères de substitution “*” et “?” représentent un moyen très efficace pour gérer des ensembles importants de fichiers, à l'aide d'un minimum de commandes. Il faut cependant garder à l'esprit, qu'une commande de suppression de fichiers par exemple, peut devenir particulièrement dangereuse si vous employez de manière abusive les caractères de substitution. Par exemple, la commande

“—> CP,SF,MD0.*—* <rc>”

supprime TOUS les fichiers de l'unité de disque fixe “MD0”. Pour vous assurer que la sélection des fichiers faite par votre commande est conforme à ce que vous pensez, il est prudent d'exécuter une commande de catalogue (utilitaire “/”) avant d'utiliser la commande de suppression, avec la même désignation de groupe de fichiers et de vérifier la liste obtenue.

Remarque :

L'usage **simultané** des caractères “*” et “?” dans une désignation de fichier, est interdit.

Ce paragraphe vous a décrit les principes généraux de mise en œuvre des caractères de substitution dans les désignations de fichiers. Cependant, chacun des utilitaires de PROLOGUE définit ses propres règles d'utilisation de ces caractères, de manière plus ou moins restrictive. Par exemple, leur usage au niveau de l'éditeur de textes EDV n'apparaît pas très fondé, car vous ne pouvez éditer qu'un fichier à la fois. Par contre, leur utilisation est fréquente avec l'utilitaire CP, pour dupliquer ou supprimer un groupe de fichiers.

La description de chacun des utilitaires de PROLOGUE, qui est faite au chapitre 4 de ce manuel, indique les possibilités d'usage des caractères de substitution admises par chacun.

2.2.2.4. Gestion de la date de création/mise à jour

Lorsqu'un fichier est créé par PROLOGUE, la date courante du système est recopiée dans le descripteur du fichier, sur le catalogue du volume. La date comporte le jour, le mois et l'année. C'est donc la date de création du fichier.

Par la suite, chaque fois qu'une opération d'écriture ou de mise à jour y est faite, cette date est entretenue. C'est donc la date de dernière mise à jour du fichier.

Une opération de lecture n'affecte pas cette date.

La date de création/mise à jour d'un fichier peut être visualisée par la commande "/" (catalogue), avec l'option GR (chapitre 4).

Remarques :

Lors de la mise à jour de la date associée à un fichier, la date de référence utilisée est la date SYSTEME, entretenue par PROLOGUE. Pour qu'elle soit significative, il est impératif, avant de démarrer votre exploitation, d'installer correctement la date SYSTEME à l'aide de l'utilitaire DATE.

Un fichier ne possède une date de création/mise à jour que si le volume qui le supporte a un indicatif A0 (ou supérieur à A0).

D'autre part, pour qu'un fichier ait une date significative, il faut qu'il ait été écrit ou modifié par un système dont la version est au moins égale à 2.2 ; en effet, les systèmes précédents ne géraient pas cette date.

2.2.2.5. Attributs de création d'un fichier

Aucun attribut de création.

2.2.2.6. Compatibilité des volumes PROLOGUE

Les versions successives de PROLOGUE sur les microprocesseurs 8 bits et 16 bits ont apporté des évolutions sur la structure des volumes. Ces évolutions ont toujours respecté le principe de la compatibilité ASCENDANTE, c'est-à-dire qu'une version de PROLOGUE est toujours capable de travailler avec une structure de volume obtenue à partir d'une version précédente. par contre, si de nouvelles fonctionnalités apparaissent, il n'est en général pas possible de les utiliser correctement avec une version antérieure. La compatibilité DESCENDANTE n'est donc pas garantie.

Dans certains cas, il est cependant possible de garantir une compatibilité uniforme dans les deux sens, par exemple, lorsque la même application est exécutée sur un parc hétérogène de machines supportant des versions différentes de PROLOGUE. Dans ce cas, PROLOGUE offre le moyen de respecter et même de garantir cette double compatibilité.

Voici la liste des différentes structures de volume définies par PROLOGUE successivement et les moyens pour respecter la compatibilité de l'une à l'autre :

Structure 8 bits

L'indicatif est 9F (notation hexadécimale).

Les noms des fichiers ne sont composés que par une longueur de nom de fichier de sept caractères et par une longueur de suffixe d'un caractère (terminologie : nom 7/1).

La date de création/mise à jour n'existe pas sur ces supports.

Structure 16 bits

L'indicatif est A0 ou A1 (notation hexadécimale).

Les désignations des fichiers peuvent avoir une longueur de nom de fichier de huit caractères et une longueur de suffixe de trois caractères (terminologie : nom 8/3). Ce type de structure gère aussi bien les fichiers 7/1 et les fichiers 8/3.

Passage d'une structure vers une autre :

Tout volume N.G.F. créé par l'utilitaire CP a pour indicatif A0, sauf si l'option "TYPE = 9F" a été choisie (dans ce cas la date n'est pas gérée). C'est lorsqu'un fichier passera à plus de 18 blocs d'extension que son support passera de l'indicatif A0 à A1. Mais une fois cet indicatif atteint, la suppression de tous les fichiers ayant plus de 18 blocs d'extension ne fera pas revenir à l'indicatif A0.

Tout volume ayant comme indicatif A1, ne peut être utilisé avec une version de PROLOGUE antérieure à la version 2.3.

D'autre part, il est possible, pour des raisons de communication avec des sites annexes, de refuser le passage des volumes de A0 en A1 : il suffit de configurer le système pour interdire ce basculement (voir le paragraphe 7.3.4. portant sur le configurateur du système).

Remarque :

La compatibilité de la structure des volumes est une compatibilité de type LOGIQUE, garantie par PROLOGUE. Par contre, la compatibilité PHYSIQUE des supports utilisés est une caractéristique liée au matériel. Pour profiter de la compatibilité des volumes il est évidemment indispensable que la compatibilité physique des supports soit respectée.

2.2.3. L'accès aux fichiers

Le Noyau de Gestion des Fichiers (N.G.F.) de PROLOGUE assure l'organisation des fichiers sur les unités de disque. Il permet la gestion élémentaire de ces fichiers à l'aide des fonctions de création, suppression, ouverture, fermeture, lecture. Le N.G.F. assure donc l'organisation de base des fichiers de PROLOGUE, sur laquelle s'appuieront tous les programmes exécutés et garantit ainsi l'homogénéité de cette organisation d'un support disque à l'autre et d'une machine à l'autre. Cependant, cette organisation élémentaire assurée par le N.G.F. n'est pas suffisante pour les besoins réels des applications qui seront utilisés. Pour pouvoir accéder aux données que vous allez gérer, vous aurez besoin par exemple, de pouvoir y accéder d'une manière sélective, suivant un certain critère, par exemple le nom du client. Pour répondre à ces besoins, qui sont des besoins fondamentaux des applications de gestion, PROLOGUE permet de définir des structures particulières de fichiers, gérées chacune par un module spécialisé qui en assure la structure et traite les accès à ces fichiers.

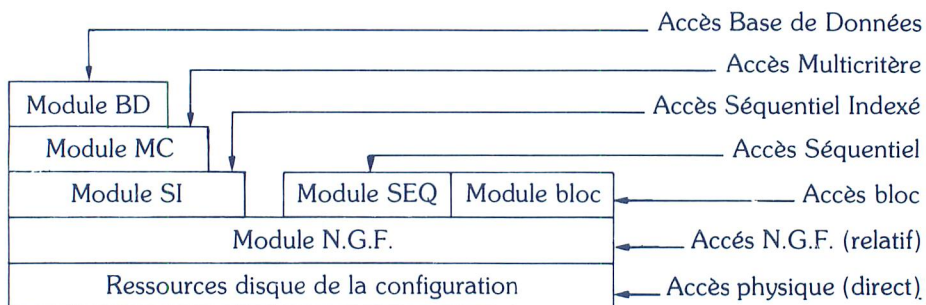
Ces modules sont appelés des Méthodes d'Accès (M.A.).

Prologue définit les méthodes d'accès suivantes :

- Accès relatif géré directement par le N.G.F. lui-même.
- Accès séquentiel géré par le module "SEQ", inclus dans le noyau système.
- Accès séquentiel indexé géré par le module optionnel "SI".
- Accès multicritère géré par le module optionnel "MC".
- Accès de base de données géré par le module optionnel "BD".

L'accès relatif et l'accès séquentiel étant gérés directement par le noyau PROLOGUE, sont toujours disponibles dans une configuration PROLOGUE. Les accès Séquentiel Indexé, Multicritère et Base de Données sont optionnels et doivent être introduits par la déclaration des modules correspondant dans le fichier de configuration "SYSCONF-S", suivent une syntaxe précisée dans le chapitre 7 de ce manuel, ou en utilisant la configuration (voir le Guide d'Installation de PROLOGUE sur votre micro-ordinateur).

La structure hiérarchique des Méthodes d'accès est la suivante :



On voit clairement, à l'aide de ce schéma, que l'accès Base de Données est le plus évolué et qu'il s'appuie sur l'accès Multicritère. Un fichier BD est une combinaison de fichiers MC.

De même, l'accès multicritère s'appuie sur l'accès Séquentiel Indexé. Un fichier MC est une combinaison de fichiers SI. L'accès Séquentiel Indexé s'appuie lui directement, sur le noyau N.G.F. Un fichier SI est donc une combinaison de fichiers N.G.F.

La méthode d'accès Séquentielle est elle, indépendante. La méthode d'accès bloc n'est qu'un complément du N.G.F. et est directement gérée par le N.G.F. tout comme l'accès séquentiel. L'accès physique représente un accès direct à la ressource disque, sans utiliser le N.G.F. Ce type d'accès n'est pas d'un usage courant et n'est mis en œuvre que pour certains besoins particuliers.

2.2.3.1. L'accès bloc

Un fichier à accès bloc se compose d'une succession d'enregistrements de longueur fixe, auxquels il est possible d'accéder directement par référence au numéro de l'enregistrement. La longueur des enregistrements est au maximum de 65535 caractères.

Un fichier à accès bloc correspond à un fichier N.G.F. dans le catalogue du volume PROLOGUE. Son suffixe est quelconque.

2.2.3.2. L'accès séquentiel

Un fichier à accès séquentiel se compose d'une suite d'enregistrements de longueur constante définie à la création du fichier séquentiel et d'un pointeur d'accès qui indique l'enregistrement courant "visible". A l'ouverture de ce fichier, le pointeur est positionné sur le premier enregistrement du fichier. Une opération de lecture fournit le contenu de l'enregistrement courant et positionne le pointeur sur l'enregistrement suivant. Une opération d'écriture correspond toujours à l'ajout d'un enregistrement en dernière position dans le fichier. Il est également possible de modifier l'enregistrement courant et de se positionner à l'enregistrement précédent.

Un fichier de type séquentiel correspond à un fichier N.G.F. dans le catalogue du volume PROLOGUE. Le suffixe d'un fichier séquentiel est quelconque.

2.2.3.3. L'accès séquentiel indexé

Un fichier séquentiel indexé est formé d'un nombre quelconque d'enregistrements limité à un maximum de 16 Mo. Chaque enregistrement est composé d'une clé qui l'identifie de manière unique et des données associées à cette clé.

A titre d'exemple, vous pouvez comparer un fichier séquentiel indexé à un dictionnaire.

Pour rechercher la signification du mot FICHIER, vous faites un accès à la section contenant les mots commençant par F, un autre accès à la page identifiée par FIC ou FICH, et un troisième accès à l'entrée précise de cette page qui définit la signification du mot FICHIER. Le mot FICHIER est une clé sans ambiguïté des données associées (définitions et exemples du mot FICHIER).

Un fichier séquentiel indexé fonctionne d'une façon analogue. On spécifie une clé d'enregistrement pour chaque article stocké par le système. C'est un identificateur de 2 à 51 octets dont la valeur est exclusive à chaque article dans le fichier. Ce peut être un numéro d'article, un numéro de facture, un numéro de pièce, un nom de client, une quelconque combinaison de données — tout ce que vous pouvez choisir pour identifier de manière unique chaque enregistrements.

Ces clés sont organisées et stockées par ordre croissant selon leur valeur hexadécimale ASCII (voir la table en Annexe A). Dans le Système de Gestion de Fichiers, lorsque vous voulez extraire un article, il suffit de spécifier la clé, et le système la recherche pour aller directement lire l'enregistrement associé. Ce qui n'exclut pas un accès séquentiel dans l'ordre des clés.

• Caractéristiques d'un Fichier Séquentiel Indexé

A la création d'un fichier séquentiel indexé, les caractéristiques suivantes sont définies :

1 — Type de fichier — Le type de fichier est défini par l'une des options :

- Clé d'enregistrement ASCII cadrée à gauche
- Clé d'enregistrement ASCII cadrée à droite
- Clé d'enregistrement binaire non cadrée

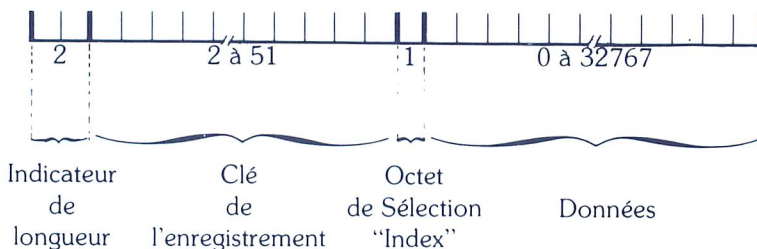
2 — Longueur des données — Longueur des articles de données à stocker ou longueur variable des données.

3 — Longueur de la clé — Toutes les clés doivent être déclarées de la même longueur, de 2 à 51 octets.

• Caractéristiques des Enregistrements Séquentiels Indexés

Une fois le fichier séquentiel indexé créé, vous pouvez insérer, lire, modifier et supprimer des enregistrements individuellement. Les enregistrements ont le format suivant :

Nombre d'octets :



où :

Indicateur de longueur	Facultatif : en cas d'utilisation, donne la longueur des données contenues dans un enregistrement.
Clé	2 à 51 octets qui identifient de manière unique cet enregistrement.
Octet de sélection "Index"	Un octet qui permet à l'utilisateur de diviser un fichier en 8 sous-niveaux. S'il n'est pas spécifié, la valeur 1 est prise par défaut en B.A.L.
Enregistrement associé	Ce sont les données associées à la clé de l'enregistrement.

• Indicateur de longueur

Il donne la longueur des données pour des enregistrements de longueur variables.

• Clé d'enregistrement

La clé d'enregistrement sert à désigner de manière unique un article de données dans un fichier séquentiel indexé. Chaque clé a une longueur de 2 à 51 octets, dans un fichier donné, toutes les clés ont la même longueur et deux clés ne peuvent pas avoir la même valeur.

Les clés et les index sont stockées dans un fichier séparé (fichier de type -I généré par le Système de Gestion des Fichiers) et sont classées, de façon que la clé de valeur la plus faible apparaisse en tête. Les clés pointent chacune vers leurs données associées et servent de table de recherche rapide pour localiser les articles de données.

Les clés peuvent être formées de tout type de données numériques ou ASCII.

À la création d'un fichier indexé, son type est précisé comme étant avec la clé ASCII cadrée à gauche, avec clé ASCII cadrée à droite ou avec clé binaire non cadrée. La sélection du cadrage correct des clés est importante.

— Clé d'enregistrement cadrée à gauche.

La clé est supposée être une chaîne de caractères ASCII. Une chaîne qui contient des blancs est lue en décalant et en éliminant les blancs jusqu'à ce que le premier caractère soit non blanc.

Exemple :

Supposons que les caractères suivants soient entrés dans une zone destinée à une chaîne de 10 caractères :

bbTATA7890
ABCDEFGHIJ
AVDCFbbbb
! () 9045bbb
123bbbbbbb

(b indique un blanc)

Comme mentionnée dans ce qui précède, toutes les clés sont dans l'ordre alphabétique et stockées sur le volume d'une façon ordonnée. Les clés sont classées selon leur code ASCII, en commençant par le caractère le plus à gauche, puis stockées avec la clé de plus faible valeur alphanumérique la première. L'évaluation d'une clé est faite de gauche à droite, selon le classement ASCII montré dans le tableau ci-dessous (voir Annexe A pour une table complète des codes ASCII). Vous voyez que beaucoup de caractères spéciaux, mais pas tous, ont une valeur plus faible que les nombres, qui ont une valeur plus faible que les lettres (le code /30 du zéro est inférieur au code /41 du A).

Classement par valeur des caractères dans le code ASCII

```

b ! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ \ ] ^ _

```

— Clé d'enregistrement cadrée à droite

La clé doit être une chaîne de caractères ASCII. Une chaîne qui contient des blancs de remplissage est lue en décalant et en éliminant les blancs, jusqu'à ce que le caractère le plus à droite soit non blanc.

Ce type de clé est surtout utilisé lorsque des chaînes ASCII numériques de différentes longueurs sont introduites comme clés. Dans la comparaison des nombres, l'évaluation a toujours lieu de gauche à droite, mais les clés sont cadrées à droite, et non à gauche. Le cadrage à droite des clés force une évaluation en tant que nombre ASCII. Considérez, par exemple, les chaînes ASCII de nombres :

```

001
02
007
00200
000300

```

Les chaînes ci-dessus sont présentées dans l'ordre où elles sont stockées lorsqu'il y a décalage à droite. Lorsque vous les imprimez en séquence, elles apparaissent dans cet ordre.

Si vous utilisez les mêmes chaînes de caractères comme clés cadrées à gauche (en entrant des chaînes avec des blancs de remplissage), elles sont cadrées à gauche et évaluées selon leur code ASCII, et apparaissent dans l'ordre suivant, après classement :

000300

001

00200

007

02

Vous remarquez que l'ordre des nombres a changé. Si vous voulez lister séquentiellement ces clés, elles apparaîtront dans l'ordre que vous venez de voir.

— Clés non cadrées (Binaire)

Dans ce type de fichier, les clés ne sont pas cadrées, mais apparaissent comme vous les avez créées. Ce type de clé est généralement utilisé lorsque les données binaires ou hexadécimales, ou une combinaison de données non ASCII sont utilisées comme clés d'enregistrement.

On peut remarquer que toute donnée identifiant un enregistrement de manière unique est utilisable comme clé, mais la définition des clés doit être pensée avec soin.

• Octet de sélection ("Index")

Un octet est associé à chaque clé dans un fichier séquentiel indexé. Cet octet de sélection permet à l'utilisateur de partager les articles d'un fichier en 1 à 8 sous-ensembles, chacun des huit bits de l'octet définissant un sous-ensemble, selon le tableau ci-dessous.

Sous-ensemble	Valeur de l'octet de sélection	
	Décimal	Binaire
1	1	0000 0001
2	2	0000 0010
3	4	0000 0100
4	8	0000 1000
5	16	0001 0000
6	32	0010 0000
7	64	0100 0000
8	128	1000 0000

Chaque article d'un fichier séquentiel indexé peut recevoir plusieurs sous-niveaux, mais doit appartenir à au moins l'un d'eux. Si l'octet de sélection n'a pas d'importance dans le programme, vous lui donnerez 1 qui est la valeur par défaut en B.A.L.

L'octet de sélection permet de diviser un fichier en plusieurs sous-fichiers. Lorsque vous recherchez un article dans le fichier, cet article est identifié par sa clé et son octet de sélection, et au moins l'un des sous-niveaux de l'octet de sélection enregistré avec l'article doit correspondre à l'un des sous-niveaux d'octet de sélection de recherche pour que la recherche aboutisse à un résultat positif.

Exemple :

Supposons qu'on ait un fichier de stock dans lequel les articles ont pour clé le code produit. A l'aide de l'octet de sélection, on peut les diviser en groupes selon un critère comme, par exemple, le délai de fabrication, etc...

Délai	Sous-ensemble (bit)	Valeur de l'octet	
		Décimale	Binaire
Disponible	1	1	00000001
30 jours	2	2	00000010
60 jours	3	4	00000100
90 jours	4	8	00001000
Article critique	8	128	10000000

Ainsi, un octet de sélection de 8 définit un article demandant un délai de 90 jours. Un octet d'état de 136 définit un article demandant un délai de 90 jours et dont le statut a en outre été défini comme critique.

• **Données**

C'est un ensemble de zones élémentaires associées à une clé définie.

La longueur, fixe ou variable, est comprise entre 0 et 32767 octets en ASCII et/ou en binaire, et/ou en numérique.

2.2.3.4. L'accès multi-critères

La méthode d'accès multi-critères est une extension de l'accès séquentiel indexé.

Elle permet l'interrogation d'un fichier de données et de mettre à jour, en temps réel ou différé, les fichiers de données. Elle donne, en outre, des informations statistiques sur l'état des fichiers.

• Caractéristiques d'un fichier multi-critères

Un fichier multi-critères se compose de quatre fichiers :

- | | | |
|--|--------------|-----------------------------|
| – Un fichier des clés | (suffixe –I) | Max. 40 octets |
| – Un fichier des données | (suffixe –D) | } Long. max.
2700 octets |
| – Un fichier de description des rubriques,
des liens et des valeurs | (suffixe –V) | |
| – Un fichier des références | (suffixe –R) | |

• Fonctions du multi-critères

Il permet d'interroger le fichier selon les valeurs de certaines zones de données, avec jusqu'à 10 critères dans une même question.

Exemple :

Ville = "PARIS", AGE (20,30)

Signifie :

Quels sont les habitants de Paris qui ont entre 20 et 30 ans ? (Il y a deux critères dans cette question).

• Définitions relatives au multi-critères

Les données associées à une clé se décomposent en RUBRIQUES.

Une rubrique a une VALEUR par article. Il peut y avoir homonymie entre valeurs.

Une valeur peut avoir une, ou plusieurs, REFERENCES qui sont les clés du séquentiel indexé.

La notion de valeur peut être généralisée.

Une liste de rubriques a une VALEUR par article. Il peut y avoir homonymie entre ces valeurs. Cette valeur multi-critère est la valeur du LIEN créé entre les rubriques. La VALEUR d'un LIEN dépend de l'ordre des rubriques.

Cette valeur obtenue par un LIEN entre des RUBRIQUES possède une ou plusieurs REFERENCES.

Un LIEN peut être constitué par la concaténation de 10 rubriques. La valeur ainsi obtenue peut avoir au maximum 40 caractères (voir Manuel B.A.L. 40 référence 05F29664).

Un LIEN est une clé secondaire.

Un CRITERE est une rubrique déclarée ; autrement dit, un CRITERE est une RUBRIQUE interrogeable.

C'est une SELECTION.

Exemples :

Fichier client : Structure de l'enregistrement séquentiel indexé

Clé=N° client	Données=Nom	Ville	Code Post.	Solde
000012	IRMA	NICE	06000	143
000037	EMILE	PARIS	75000	6742
000042	HERVE	PARIS	75000	730

Le **lien** Nom—Ville a pour valeur : IRMANICE
EMILEPARIS
HERVEPARIS

Le **lien** Ville—Nom a pour valeur : NICEIRMA
PARISEMILE
PARISHERVE

Ces liens portent sur deux rubriques et ont des longueurs de dix caractères.

• Dictionnaire des données

Le dictionnaire des données se décompose en description des rubriques et des liens et en description des références associées aux valeurs multi-critères (fichiers de suffixe V et R).

— Description des rubriques

Une rubrique se caractérise par :

- Un identificateur de rubrique (nom)
- Une longueur de rubrique (nombre d'octets)
- Un type de rubrique : alphanumérique ou binaire. (Et éventuellement le nombre de chiffres décimaux après la virgule).

— Description des liens

Un lien est une liste ordonnée d'identificateurs de rubriques.

Remarques :

Description de rubriques :

10	4	SOLDE	0004		16	19
10	1	NOM	0006	00	0	05

Où :

- | | |
|---|-------------------------|
| 1 | Code fonction |
| 2 | Numéro de rubrique |
| 3 | Identificateur |
| 4 | Longueur de la rubrique |
| 5 | Type de la rubrique |
| 6 | Position |

Description d'un lien : lien solde

20	1	4		0	4		1
----	---	---	-------	---	---	-------	---

Où :

- 1 Code fonction
- 2 Nombre de rubriques dans le lien
- 3 Liste ordonnée des numéros de rubriques
- 4 Liste des numéros de rubriques dans l'ordre du lien
- 5 Numéro de lien

Description de la valeur : solde

30	1	143	1	1
30	1	730	2	1
30	1	6742	3	1

Où :

- 1 Code fonction
- 2 Numéro de lien
- 3 Valeur
- 4 Numéro de référence
- 5 Nombre d'apparitions de cette valeur
- 6 Zone de statistiques

Description d'une référence : solde

40	1	1	12
----	---	---	----

Où :

- 1 Code de fonction
- 2 Numéro de lien
- 3 Numéro de référence
- 4 Référence = clé du séquentiel indexé

Exemple :

La rubrique 4 a pour valeur : 143
pour référence : 1
pour clé : 12

Exemples :

Rubriques

Exemples :

Dans le fichier précédemment défini on peut décrire les rubriques et les liens comme suit :

Identificateur	Longueur	Type
Nom	6	Alphanumérique
Ville	6	Alphanumérique
Code Postal	5	Alphanumérique
Solde	4	Binaire

Liens

Lien	Option
Solde	Unique
Ville-Code Postal-Nom	Ensemble minimum

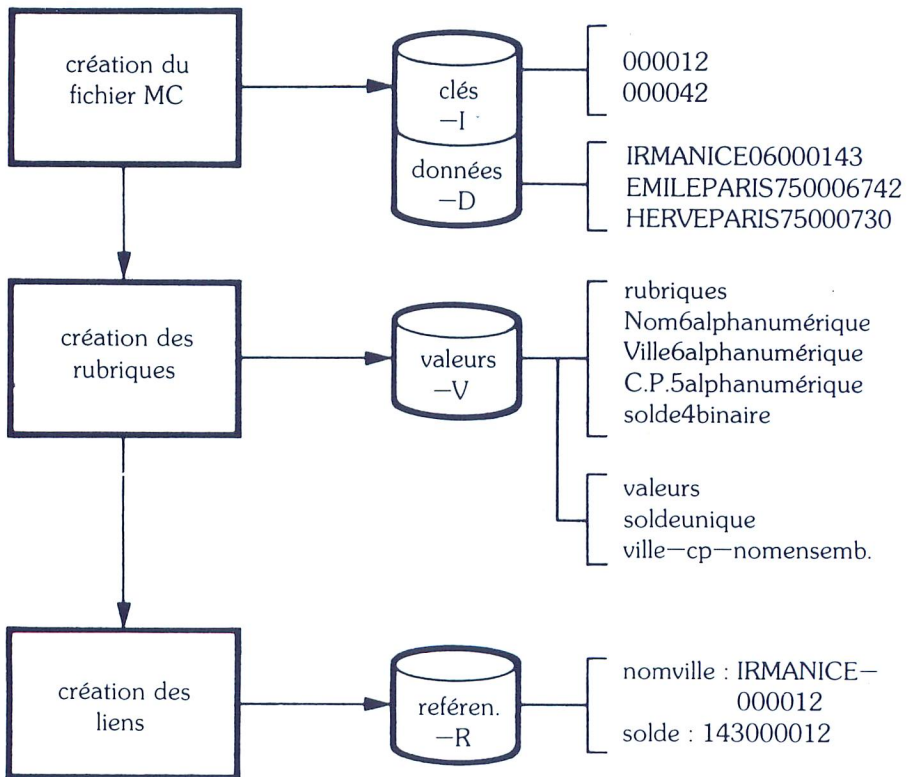
• Exploitation du dictionnaire

La recherche d'informations peut se décomposer en une succession de dénombrements pour affiner la question ou d'interrogations pour obtenir la liste des réponses.

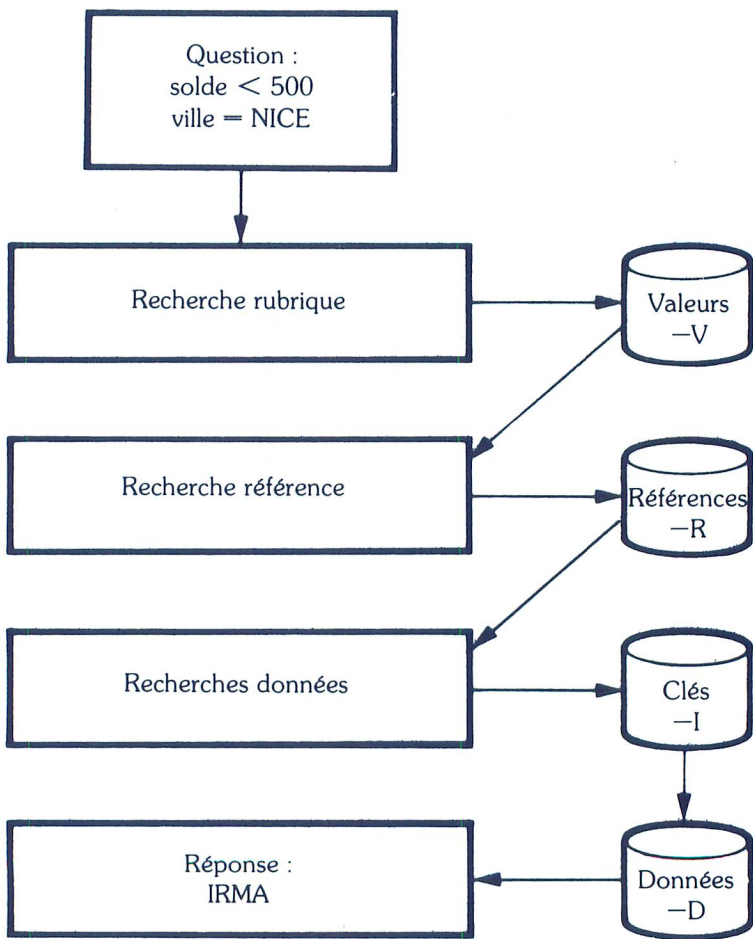
Notez que la structure multi-critère ordonnée rend inutile la fonction de tri, les réponses étant triées naturellement au moment de la création des liens.

• Fonctionnement du multi-critères

— Création



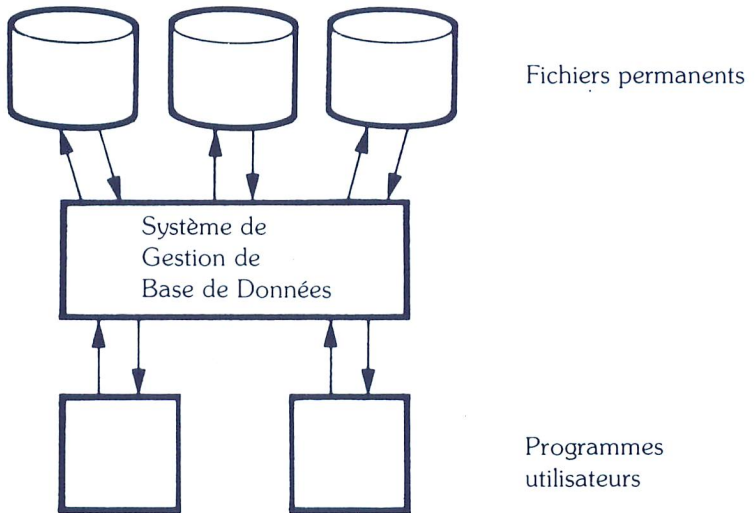
— Interrogation



2.2.3.5. Les Base de Données

PROLOGUE permet de gérer un Système de Gestion de Bases de Données Relationnelles.

— Principe de fonctionnement



Dans le multi-critères, toutes les réponses sont dans un seul et même fichier.

Une base de données est une méthode qui permet le même genre de questions que le multi-critères mais la question peut porter (pour 10 critères) sur 1 à 10 fichiers et la base de données va chercher elle-même les réponses sans que vous ayez à dire sur quels fichiers les réponses se trouvent.

Une base de données est organisée autour d'un dictionnaire (nom de la base donné par l'utilisateur) et des relations qui existent entre les données. Toutes les informations sont classées dans la base de manière à pouvoir les retrouver facilement et rapidement. Pour interroger la base, il suffit de décrire le résultat sans indiquer les accès pour y parvenir.

C'est le principe de la **base de données relationnelles**. La base de données est un ensemble de fichiers (au maximum 255) et d'articles (enregistrements logiques) comprenant des rubriques. La description des éléments qui composent cette base est centralisée dans le dictionnaire.

La base peut être consultée et modifiée à tout moment.

La base de données s'appuie entièrement sur la méthode d'accès multi-critères dont elle reprend la totalité des fonctions (voir § 2.2.3.4.), avec des données de longueur fixe.

• Définitions sur la base données

Le modèle relationnel consiste à définir chaque donnée élémentaire par son nom et l'ensemble des valeurs qu'elle peut prendre. Cela conduit à représenter les données de la base sous forme de tables appelées RELATIONS.

Chaque colonne d'une table représente une donnée et s'appelle un ATTRIBUT de la relation. Une LIGNE de la table est constituée par l'ensemble des valeurs respectives de chaque attribut.

Exemple :

Rubrique 1	Rubrique 2	Rubrique 3	
Attribut 1	Attribut 2	Attribut 3	Ligne 1 = article 1
Attribut 1	Attribut 2	Attribut 3	ligne 2 = article 2
Attribut 1	Attribut 2	Attribut 3	Ligne 3 = article 3

1 fichier multi-critères = 1 relation

• Généralités

Chaque donnée élémentaire, ou RUBRIQUE, est définie par :

son nom

son type.

Ces données sont liées entre elles par des relations qui définissent des sous-ensembles.

Remarque :

Physiquement, une relation est implémentée sous la forme d'un fichier. Afin d'être en accord avec le langage introduit par le multi-critère, nous désignerons :

Une relation par FICHER

Un attribut par RUBRIQUE

Une ligne par ARTICLE

Les rubriques d'un même fichier peuvent être liées. Cette opération s'appelle un LIEN.

Exemple :

Fichiers de base de données

Fichier clients

CLI-NO	NOM	VILLE	CD-POST
0001	ROUSSEL	PARIS	75013
0002	COZIAN	BREST	29000
0003	ROUX	ORLEANS	45000
0004	LEFEBVRE	PARIS	75018

Fichier factures

FACT-NO	NO-CLI	DATE	NO-PROD	QTE
0010	0001	01.09.84	2000	150
0021	0001	03.09.84	2000	400
0021	0001	03.09.84	2050	30
0035	0004	10.09.84	2101	200
0037	0004	10.09.84	2010	25
0040	0001	11.09.84	2010	150

• Définition des opérateurs

Les opérateurs utilisés par le Système de Gestion de Bases de Données de PROLOGUE sont :

- La restriction qui est une **sélection** par rapport à un critère.

Chaque critère se compose de :

- Nom de rubrique
- Opérateur
- Valeur

- La projection qui n'utilise que certaines rubriques d'un fichier.

- La jointure qui est un lien entre des rubriques de fichiers différents.

Exemples :

- **Restriction sur le fichier Clients**

Clients habitant paris

Forme

Rubrique	Opérateur	Valeur
CLI-NO	=	PARIS

Réponse :

0001	ROUSSEL	PARIS 75013
0004	LEFEBVRE	PARIS 75018

- **Projection sur le fichier Clients**

Quels sont les numéros et les noms des clients qui habitent Paris

Forme

VILLE = PARIS

Réponse :

0001	ROUSSEL
0004	LEFEBVRE

• Jointure entre les fichiers Clients et Factures

Clients ayant été facturés

Forme

Rubrique	Opérateur	Rubrique
CLI—NO	=	NO—CLI

Réponse :

0001	ROUSSEL	0010	01.09.84	2000	150
0001	ROUSSEL	0021	03.09.84	2000	400
0001	ROUSSEL	0021	03.09.84	2050	30
0004	LEFEBVRE	0035	10.09.84	2101	200
0004	LEFEBVRE	0037	10.09.84	2010	25
0001	ROUSSEL	0040	11.09.84	2010	150

• Le dictionnaire

Une base de données doit être identifiée par un nom. Ce nom est celui d'un fichier appelé dictionnaire, dans lequel tous les objets de la base sont enregistrés.

Ces objets sont de trois types : fichiers
rubriques
jointures.

Exemple :

Nom de la base :	BDPROD
Nom des fichiers :	CLIENTS FACTURES
Liste des rubriques :	CLI—NO NOM VILLE CO—POSTE FACT—NO NO—CLI DATE NO—PROD QTE

Liste des jointures : CLI—FAC : NO—CLI = CLI—NO

- **Fonctionnement de la base de données**

Description des fichiers de la base

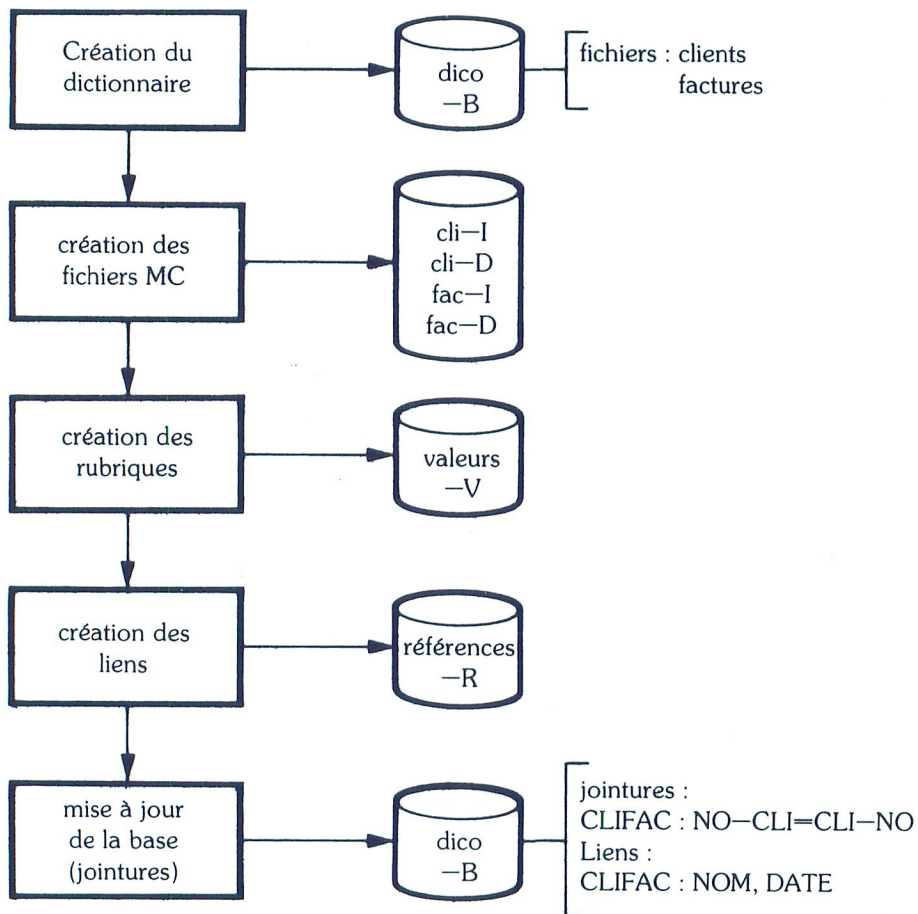
Fichier Clients

CLI-NO	NOM	VILLE	CODE POSTAL
0001	ROUSSEL	PARIS	75013
0002	COZIAN	BREST	29000
0003	ROUX	ORLEANS	45000
0004	LEFEBVRE	PARIS	75018

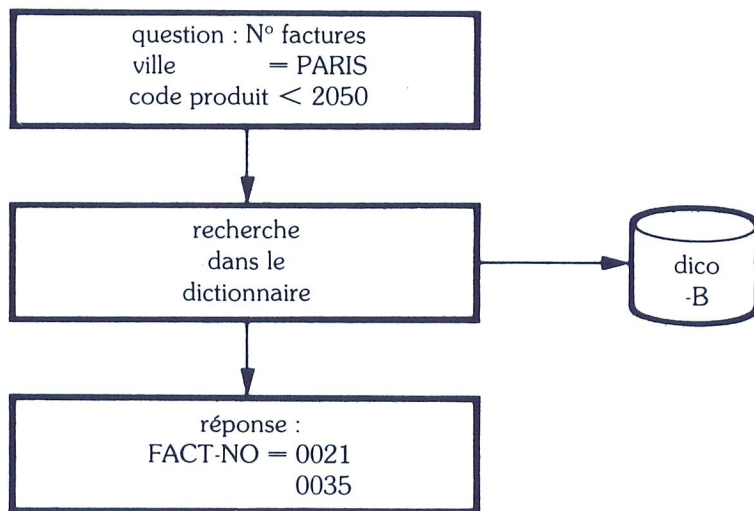
Fichier Factures

FACT-NO	NO-CLI	DATE	NO-PROD	QTE
0010	0001	01.06.83	2000	150
0021	0001	03.06.83	2000	400
0021	0001	03.06.83	2050	30
0035	0004	10.06.83	2010	200
0037	0004	10.06.83	2101	25
0040	0001	11.06.83	2010	130

Création



Interrogation





3.	DIALOGUER AVEC PROLOGUE	3-1
3.1.	Introduction	3-1
3.2.	L'interpréteur de commandes	3-2
3.2.1.	Principe de fonctionnement	3-2
3.2.2.	Utilisation du clavier	3-4
3.2.3.	Syntaxe des commandes	3-5
3.2.4.	Sélection du suffixe d'un programme	3-10
3.2.5.	Codes et messages d'erreurs	3-16
3.3.	Opérations diverses	3-20
3.3.1.	Affichage d'une bannière d'identification	3-20
3.3.2.	Exécution différée d'un programme	3-23
3.3.3.	Affichage du contexte de poste	3-24
3.3.4.	Installation de commandes résidentes	3-25
3.3.5.	Exécution de commandes par B.A.L. ou par la commande A.S.G.	3-27
3.3.6.	Les commandes et la sécurité	3-28

3.DIALOGUER AVEC PROLOGUE

3.1. INTRODUCTION

Après avoir chargé le système PROLOGUE, vous remarquez sur votre écran le symbole “—>” qui indique que le système est en attente d’une commande à exécuter. Les commandes sont transmises au système par l’intermédiaire du clavier. L’écran et le clavier, qui ensemble forment la console, sont le moyen par lequel vous “dialoguez” avec Prologue. Dans une configuration multiposte, chaque utilisateur possède une console qui lui permet de dialoguer d’une façon indépendante avec le système. Le symbole “—>” est appelé le signal d’appel du système. Il est affiché par un module particulier appelé “Interpréteur de commandes”. Ce chapitre vous indique le fonctionnement de l’interpréteur de commandes de PROLOGUE.

3.2. L'INTERPRETEUR DE COMMANDES

3.2.1. Principe de fonctionnement

L'interpréteur de commandes est un programme particulier dont le rôle est de vous permettre d'exécuter tous les programmes dont vous aurez besoin lors de votre exploitation, aussi bien les programmes utilitaires de PROLOGUE que les programmes d'application.

Un programme est un fichier, catalogué sur un support disque, composé d'instructions exécutables par le système.

Il existe plusieurs types de programmes exécutables par PROLOGUE. Le type d'un programme est identifié par le suffixe du fichier correspondant. Lorsque le nom d'un programme est donné dans une commande, il n'est en général pas nécessaire de donner son suffixe. En effet, l'interpréteur de commandes détermine automatiquement le suffixe du fichier à charger en mémoire.

Le paragraphe 3.2.4. décrit les différents suffixes de programmes exécutables par PROLOGUE et la façon dont l'interpréteur de commandes sélectionne le type du fichier correspondant.

Exécuter un programme consiste donc à charger le fichier correspondant en mémoire centrale et à l'exécuter en suivant les paramètres de la commande d'appel.

Le fonctionnement de l'interpréteur de commandes se déroule suivant les étapes ci-dessous :

- 1 — Analyse de la commande frappée au clavier par l'utilisateur.
- 2 — Recherche du fichier sur disque suivant un suffixe défini.
- 3 — Chargement de ce fichier en mémoire centrale.
- 4 — Exécution du programme en suivant les paramètres fournis dans la commande d'appel.
- 5 — En fin d'exécution, suppression du programme de la mémoire centrale.
- 6 — Fermeture de tous les fichiers restés éventuellement ouverts.
- 7 — Affichage du signal d'appel "—>" et attente d'une nouvelle commande.

Lorsque le signal d'appel est affiché, le système attend votre commande et est donc inactif. Cette situation est appelée "point de repos" du système. A ce niveau, vous êtes assurés que tous les fichiers utilisés sont fermés et que toutes les données mises à jour sont cohérentes et correctement mémorisées. Vous pouvez donc sans risque procéder au changement des volumes amovibles ou terminer votre activité suivant les indications du paragraphe 1.2.7.

Si la configuration utilisée est multiposte, il faut s'assurer que tous les postes sont au point de repos avant de procéder à un changement de volume.

De même, si l'utilitaire de SPOUL imprimante est actif, les fichiers "liste en attente d'impression" peuvent être ouverts, avec des accès en cours (voir chapitre 4).

Lorsque le programme appelé s'exécute, l'Interpréteur de commandes est inactif et c'est le programme lui-même qui a le contrôle de l'exécution. Il peut notamment, en fonction de ses besoins propres, afficher des messages sur l'écran ou attendre des données à partir du clavier.

3.2.2. UTILISATION DU CLAVIER

Il est important de bien connaître son clavier car c'est l'élément le plus utilisé pour conduire l'exploitation d'un système. Il faut pouvoir à la fois aller vite lorsque la quantité des données à introduire est importante, et à la fois être très précis dans la syntaxe des commandes. Des erreurs de frappe peuvent causer des dommages importants et vous faire perdre plusieurs jours de travail !

Il existe une grande variété de claviers, plus ou moins complets, et plus ou moins ressemblant à celui d'une machine à écrire. Ce manuel n'étant pas attaché à un modèle particulier de machine, ne peut vous en faire une description complète et précise. Nous pouvons cependant vous en présenter la structure générale ainsi que les touches ayant un rôle particulier dans les commandes PROLOGUE.

Un clavier se compose d'un bloc de touches alphanumériques permettant l'introduction de données alphabétiques ou numériques, d'un bloc de touches numériques pour l'introduction des nombres, de touches de fonctions associées à des fonctions particulières et de diverses touches de commandes prédéfinies. On distingue :

- 1 – La touche de validation ou “retour chariot”, situé à droite du bloc alphanumérique. Dans ce manuel, cette touche est symbolisée par “<rc>”. Lorsqu'une commande ou une valeur est frappée au clavier, cette touche indique la fin de la commande et permet au système de la prendre en compte.
- 2 – La touche CONTROL désignée par “CTRL” qui permet d'altérer la signification des touches alphabétiques. La touche CTRL s'emploie en combinaison avec une autre touche alphabétique.
Par exemple, la frappe de “CTRL M” a une action équivalente à la frappe de la touche “retour chariot”, les codes produits étant identiques.
L'interprétation des codes fournis par la touche de contrôle sont propres à chacun des programmes utilisés.
- 3 – Les touches de Majuscules (bloquées ou non), situées de part et d'autre de la barre d'espacement qui permettent de passer en notation majuscule.

- 4 – Les touches de déplacement du curseur qui permettent son déplacement vers le haut, le bas, à gauche et à droite et le retour du curseur en haut et à gauche de l'écran (home).
- 5 – Les touches de fonction programmables, dont le nombre est variable suivant les claviers. PROLOGUE permet d'attribuer à chaque touche de fonction un code qui peut être une chaîne de caractères quelconque. chaque fois que la touche de fonction est frappée, la chaîne de caractères correspondante est transmise au système. Ce mécanisme permet d'associer une commande complète à une touche de fonction et de déclencher la commande par la simple frappe de la touche. L'utilitaire TFP décrit dans le Guide d'Installation de PROLOGUE sur votre micro-ordinateur permet de réaliser la programmation des touches de fonction.

Lorsqu'un caractère alphabétique ou numérique est frappé au clavier, il est affiché sur la position courante du curseur qui est alors déplacé d'une position sur la droite. Le curseur est en général un tiret ou un pavé clignotant représenté dans ce manuel par le symbole“-”.

Lorsque vous frappez une commande, vous pouvez utiliser les touches de contrôle suivantes :

La touche retour chariot ou <rc> qui vous permet de valider la commande, c'est-à-dire d'en indiquer la fin et d'en permettre la prise en compte par le système. Les touches CTRL et M simultanées (CTRL M), ont une action équivalente à la touche <rc>

La touche curseur gauche (ou <-) qui vous permet de ramener le curseur sur la caractère précédent afin d'en faire la correction.

La touche effacement (ou EFF ou DEL) qui vous permet de positionner le curseur sur le caractère précédent et d'effacer ce caractère.

La touche curseur droit (ou ->) qui vous permet de déplacer le curseur vers la droite sans modification des caractères préalablement entrés.

La touche annulation ou les touches CTRL et A simultanées (CTRL A), qui permettent de ramener le curseur en début de commande.

Ces touches de contrôle vous permettent de corriger votre commande avant d'en lancer l'exécution par la touche retour chariot.

Remarque :

La frappe de la touche “CTRL + caractère” génère un code particulier qui n’est pas affichable. Si ce code ne correspond pas à une fonction de l’interpréteur de commandes, il est affiché sous la forme “ ^ C” ou “C” est le caractère frappé simultanément. Mais en interne ce code est considéré comme un caractère unique. Si vous déplacez le curseur sur ce code vers la gauche ou vers la droite, vous remarquerez qu’il se déplace de 2 positions simultanément. Il n’est pas possible de dissocier “ ^” et “C”.

Note :

Si vous avez lancé l’exécution d’une commande, la touche “—>” vous la réaffiche caractère par caractère, une fois la commande terminée.

3.2.3. SYNTAXE DES COMMANDES

La forme générale d'une commande est la suivante :

—> desfic[,liste de paramètres]<rc>

où :

“—>”

est le signal d'appel affiché par l'interpréteur de commandes.

“desfic”

est le nom du programme à exécuter, c'est-à-dire le nom du fichier correspondant. En général le suffixe de fichier est omis. Il prend donc une valeur par défaut suivant un mécanisme expliqué au paragraphe 3.2.4. Si le programme est un utilitaire de PROLOGUE, c'est un fichier de type binaire dont le suffixe est “X”.

Si la désignation de l'unité est omise, c'est l'unité implicite programme du poste qui est sélectionnée, conformément à la configuration du système. si l'unité sur laquelle se trouve le programme à exécuter n'est pas l'unité implicite programme, il est alors nécessaire de la spécifier explicitement dans la désignation du programme.

Exemple :

Supposons que l'unité implicite programme de votre poste soit l'unité “MD0”. La commande

“—>CP,param <rc>”

recherchera le fichier “CP-X” sur l'unité de disque dur “MD0”, alors que la commande

“FL0”.CP, param<rc>”

le recherchera sur l'unité de disquette “FL0”.

“Liste
de paramètres”

sont des données fournies au programme permettant de conditionner son exécution. Elles peuvent être des désignations de fichiers, des valeurs, ou des mots-clés permettant de sélectionner une option de fonctionnement. Les paramètres sont séparés par une virgule. Un paramètre peut être obligatoire ou optionnel. Lorsqu'un paramètre optionnel est omis, une valeur par défaut lui est toujours attribuée.

La nature et l'interprétation des paramètres sont spécifiques à chaque programme. Chaque utilitaire de PROLOGUE a ses propres paramètres qui sont décrits dans sa mise en œuvre (voir chapitre 4). Lorsqu'un paramètre est une désignation de fichier, si la désignation de l'unité est omise, c'est l'unité implicite fichier du poste qui est sélectionnée.

Exemple :

Si l'unité implicite fichier de votre poste est l'unité de disquette “FL1”, la commande
“—> CP,DF,data—s, FL0<rc>”
recherchera le fichier “data—s” sur l'unité de disquette “FL1”.

<rc> est le symbole de la touche retour chariot qui doit toujours terminer une commande. C'est seulement à la frappe de cette touche que la commande est prise en compte par le système, analysée et exécutée. La validation de la commande est indiquée par le déplacement du curseur d'une position à droite. Lorsque le système a fini de charger le programme en mémoire et s'apprête à l'exécuter, le curseur est ramené à sa position initiale donnant ainsi une indication sur le temps de chargement.

L'exécution correcte d'une commande est ordinairement signalée par l'apparition d'un tiret ("—") à droite de la ligne de commande, accompagnée par l'affichage du signal d'appel de l'interpréteur de commandes sur la ligne suivante :

```
—> commande <rc> —  
—> .
```

Le symbole "—" représente le curseur dont la position indique où sera affiché le prochain caractère frappé.

Donnons quelques exemples de commandes courantes que vous serez amené à utiliser fréquemment :

```
—> /,fl1,lis=lo<rc>
```

Cette commande permet d'obtenir le catalogue complet du volume installé sur l'unité de disquette "FL1". Le fichier binaire "/-X" est chargé à partir de l'unité implicite programme. Le mot-clé optionnel "LIS=L0" vous permet d'obtenir l'édition du catalogue sur l'imprimante implicite de votre poste.

L'utilitaire / (catalogue) peut toujours être employé sans aucun risque de destruction de vos volumes. En effet, il n'opère qu'en lecture et peut être utilisé sur des volumes protégés en écriture.

```
--> fl0.CP,DV,fl0,md0 <rc>
```

Cette commande recopie tous les fichiers de la disquette installée sur "FL0" sur le disque fixe de la machine. Cette commande est pratique pour recopier sur le disque un logiciel que vous avez reçu sur disquette, par exemple PROLOGUE. Attention cependant, tous les fichiers du disque dont le nom est identique à l'un des fichiers de la disquette seront supprimés et remplacés par les fichiers d'origine. Vous remarquerez que le programme "CP-X" sera chargé à partir de l'unité "FL0" ceci quel que soit l'unité implicite programme de votre poste.

—> CPS,fl0,fl1,PM<rc>

Cette commande duplique physiquement le contenu du support installé sur “FL0” vers le support installé sur “FL1”. L’option “PM” provoque un prémarquage préalable du support destination avant duplication. Cette opération permet d’obtenir une sauvegarde d’un support ou d’un volume PROLOGUE. Elle ne peut être exécutée que si les supports d’origine et destination ont des caractéristiques identiques.

ATTENTION : Toutes les données préalablement contenues sur le support destination seront perdues.

La façon de procéder à l’affichage du contexte de son poste est décrit au paragraphe 3.3.3. Le contexte de poste contient les unités implicites programme et fichier.

3.2.4. Sélection du suffixe d’un programme

Nous avons vu que lorsque vous donnez un nom de programme à exécuter, il est inutile d’indiquer explicitement le suffixe du fichier dans la commande. En effet, PROLOGUE le détermine automatiquement.

L’interpréteur de commande ne peut charger et exécuter que des fichiers de suffixe —X.

Pour simplifier la syntaxe des commandes d’appel, PROLOGUE détermine automatiquement quel fichier binaire il y a lieu de charger pour exécuter des fichiers dont le suffixe est prédéfini.

Exemple :

Un fichier traduit B.A.L. a pour suffixe "T". Ce fichier qui est le résultat de la traduction d'un fichier source B.A.L. ne peut être exécuté directement par PROLOGUE, mais par un programme binaire spécial appelé l'exécuteur B.A.L. et dont le nom est "EX-X". Normalement une commande complète pour exécuter un programme B.A.L. devrait être :

"->" EX,prog,param...<rc>"

où "prog" est le nom du programme B.A.L. de suffixe "T". Pour PROLOGUE, "prog" n'est pas directement un programme mais plutôt un paramètre d'exécution fourni à EX.

Pour simplifier la commande d'appel, PROLOGUE autorise la commande :

"->" prog,param...<rc>"

Lorsqu'il détecte que le suffixe de "prog" sur le volume est "T", Prologue procède implicitement au chargement de l'exécuteur B.A.L. et lui fournit en paramètre "prog".

La commande complète explicite reste toujours utilisable. Les deux commandes ne sont en fait pas totalement identiques, la détermination des unités implicites étant faite de façon différente. Suivant la configuration, il sera commode d'employer l'une ou l'autre des formes.

Les différents suffixes de fichiers reconnus implicitement par PROLOGUE sont les suivants, décrits par ordre de priorité :

— Suffixe "X" :

Ce sont les fichiers binaires directement exécutables par le micro-processeur. Ils sont composés d'instructions machine. La plupart des utilitaires de PROLOGUE sont de suffixe "X".

— Suffixe "T" :

Ce sont les programmes traduits B.A.L. Ils sont obtenus après traduction d'un programme source B.A.L. Ils sont exécutés par un programme spécial appelé l'exécuteur B.A.L. (EX-X). Un programme B.A.L. de suffixe "T" sera implicitement recherché par l'exécuteur B.A.L. sur l'unité implicite FICHIER.

- Suffixe “EXE” ou suffixe “COM” :
Ce sont les programmes binaires exécutables par le décor MS/DOS (fichier MSDOS-X). Un programme de ce type sera implicitement recherché par le décor MSDOS sur l'unité implicite programme.
- Suffixe “CMD” :
Ce sont les programmes binaires exécutables par le décor CP/M-86 (fichier CMP-X). Un programme de ce type est implicitement recherché par le décor CPM sur l'unité implicite PROGRAMME.
- Suffixe “ASG” :
Ce sont les fichiers source d'enchaînement de commandes, exécutables par l'utilitaire ASG (fichier ASG-X). Un programme de ce type est recherché sur l'unité implicite FICHIER.

La détermination automatique du suffixe est assurée de la façon suivante :

- 1 – PROLOGUE regarde en premier si le programme désigné est un programme résident, présent en mémoire centrale. Dans ce cas il n'y a aucun chargement et l'exécution du programme est immédiate. Le paragraphe 3.3.4. décrit l'installation et la mise en œuvre des programmes résidents.
- 2 – Sinon, PROLOGUE recherche le fichier “nomfic-X”. Si ce fichier existe, le programme appelé est un programme binaire PROLOGUE. Dans ce cas, le programme est chargé et exécuté directement.
- 3 – Sinon, PROLOGUE recherche le fichier “nomfic-T”, c'est-à-dire un programme traduit B.A.L. Si ce fichier existe, l'exécuteur B.A.L. est chargé en mémoire et exécuté avec “nomfic-T” en paramètre d'appel.

La commande

“-> nomfic<rc>”

est donc équivalente à la commande

“-> EX,nomfic<rc>”.

L'exécuteur B.A.L. (fichier EX-X), doit être soit installé en tant que programme résident, soit nomfic doit être présent sur l'implicite programme.

ATTENTION :

La commande

"-> nomfic<rc>"

est plus précisément équivalente à la commande

"->IP.ex,IF,nomfic-T <rc>",

où "IP" désigne l'unité implicite programme et "IF" désigne l'unité implicite fichier. Si ces unités sont identiques, il ne se pose aucun problème de recherche. Si elles sont différentes, le même fichier "nomfic-T" sera recherché sur l'unité implicite programme par l'interpréteur de commandes et simultanément sur l'unité implicite fichier par l'exécuteur B.A.L. Dans ce cas, il sera donc préférable d'utiliser une syntaxe explicite d'appel :

"-> EX,nomfic,...<rc>".

- 4 — Sinon PROLOGUE recherche le fichier "nomfic-EXE" ou "nomfic-COM", c'est-à-dire un programme binaire exécutable par le décor MSDOS. Si ce fichier existe, le décor MSDOS est chargé en mémoire et exécuté avec "nomfic-EXE" ou "nomfic-COM" en paramètre d'appel. La commande

"-> nomfic<rc>"

est alors équivalente à la commande

"-> MSDOS.nomfic<rc>".

Le décor MSDOS (fichier MSDOS-X), doit être présent sur la même unité que "nomfic".

- 5 — Sinon PROLOGUE recherche le fichier "nomfic-CMD", c'est-à-dire un programme binaire exécutable par le décor CP/M. Si ce fichier existe, le décor CP/M est chargé en mémoire et exécuté avec "nomfic-CMD" en paramètre d'appel. La commande

"-> nomfic<rc>"

est alors équivalente à la commande

"-> CPM,nomfic<rc>".

Le décor CP/M (fichier CPM-X), doit être présent sur la même unité que "nomfic".

- 6 – Sinon PROLOGUE recherche le fichier “nomfic-ASG”, c’est-à-dire un fichier d’enchaînement de commandes, fichier source créé avec un éditeur de textes. Si ce fichier existe, l’utilitaire ASG est chargé en mémoire et exécuté avec “nomfic-ASG” en paramètre d’appel. La commande

“—> nomfic<rc>”

est alors équivalente à la commande

“—> ASG,nomfic<rc>”.

L’utilitaire ASG (fichier ASG-X) doit être soit un programme résident, soit nomfic doit être présent sur l’implicite programme.

ATTENTION :

La commande

“—> nomfic,...<rc>”

est plus précisément équivalente à la commande

“—> IP.ASG,IF.nomfic-ASG,...<rc>”

où “IP” désigne l’unité programme et “IF” l’unité implicite fichier. Si ces unités sont identiques, il ne se pose aucun problème de recherche. Si elles sont différentes, “nomfic-ASG” sera recherché sur l’unité implicite programme par l’interpréteur de commandes et simultanément sur l’unité implicite fichier par la commande “ASG”. Dans ce cas il est préférable d’utiliser une syntaxe de commande explicite :

“—> ASG,nomfic,...<rc>”

- 7 – Si le fichier “nomfic” n’existe pas avec aucun de ces suffixes, le message d’erreur suivant est affiché :

—> nomfic<rc>

“nomfic” : programme inexistant

—> .

Remarque :

Si le suffixe du programme est donné explicitement dans la commande d'appel, le mécanisme décrit précédemment n'a pas lieu. Seul le fichier désigné explicitement est recherché.

"-> FPAIE-EXE,param<rc>"

ne pourra s'appliquer qu'au fichier "FPAIE-EXE" exclusivement.

Mais attention : si vous frappez la commande

"-> FPAIE-t,param..<rc>"

et que "FPAIE-t" est effectivement un programme traduit B.A.L., vous provoquez une erreur d'exécution fatale car PROLOGUE ne peut directement exécuter un tel type de programme.

3.2.5. Codes et messages d'erreurs

Après avoir frappé une commande et l'avoir validée, il est possible que vous obteniez une erreur d'exécution. Cette erreur se traduit par l'affichage d'un code erreur et d'un libellé explicatif et l'abandon de l'exécution de la commande. Votre écran se présente par exemple ainsi :

```
-> CP,SV,f10<rc>  
      Erreur de commande n° 30 Syntaxe incorrecte  
-> .
```

Dans ce cas, l'erreur est due à l'emploi d'un mot-clé inconnu de l'utilitaire CP (mot-clé SV). L'exécution en cours est abandonnée et le traitement entamé doit être considéré comme non significatif.

Il existe deux causes très fréquentes d'erreurs :

- Une erreur de syntaxe dans la commande d'appel, due soit à une erreur de frappe, soit à une mauvaise compréhension de la syntaxe de la commande.
- Une condition non réalisée et cependant nécessaire à l'exécution de la commande (par exemple : absence du fichier origine d'une copie).

Aussi longtemps que vous n'avez pas frappé le retour chariot qui valide votre commande, vous pouvez en vérifier et en corriger la syntaxe au moyen des touches de contrôle du curseur.

Si l'erreur est une erreur standard au système, dite erreur système, elle est affichée sous la forme standard suivante :

“Catégorie de l'erreur” n° XX “libellé erreur”

“catégorie erreur”

indique la catégorie à laquelle appartient l'erreur. On distingue les catégories suivantes :

- “Erreur périphérique” :

Ce sont les erreurs liées à une opération de lecture ou d'écriture sur un support périphérique. Ces erreurs sont spécifiques à chaque type de support. Leurs valeurs sont réparties de 1 à 15. Par exemple :

Erreur n° 2 erreur de lecture/écriture

Erreur due à une défaillance du périphérique adressé dans une opération de lecture ou d'écriture.

— “Erreur système” :

Cette catégorie regroupe les erreurs retournées par le moniteur multitâches et la gestion mémoire de PROLOGUE, ainsi que certaines erreurs dont la signification est générale. Leurs valeurs sont réparties de 20 à 29 et de 56 à 59. Par exemple :

Erreur système n° 27 débordement mémoire

Il n'y a plus assez de mémoire centrale disponible pour pouvoir exécuter la fonction demandée.

Erreur système n° 56 fonction non supportée

La fonction demandée par un programme ou une commande utilisateur requiert un élément inexistant dans la configuration système utilisée.

— “Erreur fichier” :

Ce sont toutes les erreurs retournées par le noyau de gestion des fichiers (N.G.F.) de PROLOGUE.

Elle peuvent survenir dans toutes les commandes de manipulation de fichiers ou dans les programmes d'application utilisés. Leurs valeurs sont réparties de 40 à 69. Par exemple :

Erreur fichier n° 40 fichier inexistant

Une commande utilisateur ou un programme fait référence à un fichier dont le nom n'existe pas dans le catalogue du support disque désigné.

Erreur fichier n° 46 débordement de volume

Tous les granules du volume référencé sont alloués. Il n'est donc plus possible d'allouer un granule pour un nouveau fichier ou d'en agrandir un.

Erreur fichier n° 54 format de volume incorrect

Le support présent dans l'unité de disque référencée ne contient pas un volume de format correct. Cette erreur survient le plus souvent sur les unités de disquette, lorsque vous réutilisez des disquettes ayant préalablement été formatées par un autre système d'exploitation ou sur un autre type d'unité.

— “Erreur de commande” :

Ce sont toutes les erreurs de syntaxe ou de logique introduites lors de la frappe des commandes d'appel des programmes et détectées par l'interpréteur de commandes. Leurs valeurs sont réparties de 30 à 39. Par exemple :

Erreur de commande n° 30 syntaxe incorrecte

Cette erreur assez fréquente est due le plus souvent à des erreurs de frappe dans la commande d'appel : Nom de fichier incorrect, mot-clé invalide.

Le numéro de l'erreur identifie exactement l'erreur et permet de se référer à l'annexe B de ce manuel, où l'ensemble des codes erreurs du système sont décrits de façon détaillée ainsi que les solutions que l'on peut appliquer afin d'y remédier. Le numéro d'erreur est toujours donné en décimal.

Le libellé de l'erreur est un message qui rappelle brièvement la signification du code erreur.

Les codes erreurs ont des importances très diverses en ce qui concerne leur gravité et leur fréquence d'apparition. Certaines erreurs peuvent survenir assez normalement en exploitation et ne correspondent qu'à de simples erreurs de manipulation.

D'autres codes erreurs sont de simples indicateurs signalant la présence d'une condition particulière : “Protection écriture”, “Fin de fichier”, etc...

Enfin, certains codes erreurs ne doivent pas survenir en exploitation normale et sont liés à des défaillances graves de fonctionnement du logiciel ou du matériel. Exemple : “Erreur de lecture/écriture, “n° de tâche inconnu”.

Des codes erreurs au-delà de 69 existent. Ils ne sont pas utilisés directement par le noyau PROLOGUE mais par des modules optionnels configurables. Par exemple, la méthode d'accès Base de données (module BD—X) utilise les codes erreurs de 170 à 189. La signification de ces codes est décrite dans la documentation de référence attachée au produit.

Une liste complète des codes erreurs est fournie dans l'annexe B de ce manuel.

D'autres types d'erreurs peuvent survenir lors de l'exploitation de vos programmes. Ces erreurs, ainsi que les messages qui leur sont associés, sont spécifiques à chacun des programmes utilisés.

Dans ce cas, pour comprendre la signification de l'erreur et connaître les solutions que l'on peut lui apporter, il faut se reporter au guide utilisateur du programme utilisé.

3.3. Opérations diverses

Nous allons détailler maintenant diverses opérations qu'il est possible d'exécuter avec l'interpréteur de commandes et les commandes de PROLOGUE.

3.3.1. Affichage de la bannière d'identification

La bannière d'identification d'un programme indique la nature et la fonction de ce programme, par exemple "Catalogue de volume", et la version courante utilisée. Cette bannière est donc utile pour connaître facilement la nature d'un programme apparaissant dans le catalogue d'un volume dont vous ignorez le nom et surtout pour connaître quelle en est sa version. En effet, un programme est amené à évoluer à une cadence plus ou moins élevée, afin de lui apporter des corrections ou des améliorations. Dans le cas où il semble que votre système ait un comportement anormal, il est important de s'assurer que la version des programmes utilisés est correcte.

L'affichage de la bannière est possible avec tous les programmes utilitaires de PROLOGUE. Il est possible que ce ne soit pas le cas avec tous les programmes utilisés dans votre exploitation.

Pour obtenir la bannière d'identification d'un programme, procédez de la façon suivante :

- Lorsque PROLOGUE attend votre commande, chargez en mémoire le programme en tapant son nom sans paramètres, c'est-à-dire immédiatement suivi par la touche retour chariot <rc>.

Exemple :

"—> CP<rc>

- Le curseur se déplace d'une position sur la droite pour indiquer la prise en compte de la commande et lorsque le programme est chargé, une virgule est affichée immédiatement à la suite du nom du programme.

– Le système attend alors une entrée clavier. Appuyez à nouveau sur la touche retour chariot <rc>. La bannière d'identification du programme est alors affichée sous la forme suivante :

```
—> PROG<rc>
      Libellé du programme VERSION x.y/z.
      Copyright (c) 1984 BULL Micral
—> .
```

x est l'indice majeur de la version du programme
y est l'indice mineur de la version
z est l'indicateur de mise à jour du programme

– Après affichage de la bannière, il y a retour immédiat à l'interpréteur de commandes et affichage du signal d'appel.

Si vous constatez un mauvais fonctionnement de vos programmes, il est important de noter les versions des programmes utilisés, ceci afin de mieux cerner le contexte du problème.

Ce mécanisme d'affichage de la bannière d'un programme est supporté par la plupart des programmes utilitaires de PROLOGUE. Cependant certains utilitaires présentent des différences de fonctionnement.

Les utilitaires “/” (catalogue), “DATE”, “STATUS”, “STMEM” s'exécutent immédiatement après leur chargement, même en l'absence de tout paramètre dans la commande d'appel. En conséquence, leur bannière d'identification est affichée systématiquement à chaque exécution.

Exemple :

```
—> /<rc>
      Catalogue de volumes Version 3.4a
      Copyright (c) 1984 BULL Micral
```

Vol : xxxxxxxx date: xx/xx/xx Nbfic: xxx/xx Lg.gr: xx Occup: xx/xx

```
.....
.....
.....
```

—> .

L'utilitaire "CPS" peut être chargé en mémoire sans exécution immédiate, mais lorsque la touche retour chariot <rc> est enfoncée, il y a affichage de la bannière d'identification mais le programme ne retourne pas directement à l'interpréteur de commandes. L'appui à nouveau sur <rc> permet la sortie du programme et le retour à l'interpréteur. Le déroulement est le suivant :

—> CPS<rc>,<rc>

Copie physique de supports sectorisés Version 2.3a

Copyright (c) BULL Micral

Origine : <rc>

—> .

Pour obtenir plus de précisions sur le fonctionnement de ces utilitaires, reportez-vous au chapitre 4 de ce manuel.

3.3.2. Exécution différée d'un programme

Lorsque vous frappez au clavier la commande d'appel d'un programme, l'exécution du programme a lieu immédiatement après son chargement en mémoire. Il est parfois pratique de pouvoir faire une pause entre le chargement d'un programme et son exécution. Ceci peut être obtenu d'une façon similaire à celle décrite dans le paragraphe précédent :

- Chargez le programme en mémoire en tapant le nom du programme seul, sans paramètres d'appel et suivi immédiatement par la touche retour chariot `<rc>` :
 `-> PROG<rc>`
- Lorsqu'une virgule est affichée immédiatement à la suite du nom du programme, le programme est chargé en mémoire :
 `-> PROG,-`
- Il est alors possible de frapper à la suite des paramètres d'appel de la commande d'une façon identique à un appel habituel :
 `-> PROG,paramètres...<rc>`
- la frappe de la touche retour `<rc>` . provoquera l'exécution du programme.

Ce mécanisme d'exécution différée permet de procéder au changement des volumes amovibles, par exemple des disquettes, avant exécution du programme.

La plupart des utilitaires PROLOGUE autorisent ce mécanisme. Cependant les utilitaires suivants ne permettent pas une exécution différée :

`"/` (catalogue), `"DATE"`, `"STATUS"`, `"STMEM"`

3.3.3. Affichage du contexte de poste

Lorsque le système est en attente d'une commande utilisateur, c'est-à-dire lorsque le curseur est immédiatement à la suite du signal d'appel, la frappe immédiate de la touche retour chariot <rc> provoque l'affichage du contexte de poste. Les informations suivantes sont affichées : Version courante du système PROLOGUE, numéro du poste utilisateur, ressources implicites programme, fichier, imprimante.

Exemple :

—> <rc> donne l'affichage suivant :
—> Prologue Vx.x Poste 0 Implicites : MD0,MD0, IM0
—> .

La version courante de PROLOGUE est la version x.x. Le numéro de poste est le numéro "0", c'est-à-dire le premier. L'unité de disque implicite programme est l'unité "MD0" (disque fixe). L'unité de disque implicite fichier est également "MD0". L'unité implicite d'impression est la ressource "IM0".

Autre exemple de contexte de poste :

—> Prologue Vx.x Poste 1 Implicites : FL0,FL1,--
—> .

La version courante de PROLOGUE est la version x.x. Le numéro de poste est "1". L'unité implicite programme est l'unité de disquettes "FL0". L'unité implicite fichier est l'unité de disquettes "FL1". Il n'y a pas de ressource implicite imprimante.

Remarque :

Les postes utilisateurs sont numérotés à partir de 0. Dans une configuration monoposte, le numéro de poste est donc 0. Le système PROLOGUE ayant une capacité maximum de 8 postes, les numéros de postes vont donc de 0 à 7 maximum.

3.3.4. Installation de commandes résidentes

Certains programmes utilitaires de PROLOGUE ont la possibilité d'être installés en mémoire centrale de manière permanente lors de l'initialisation du système. Cette possibilité est très intéressante pour 3 raisons :

- 1 — Cela permet de supprimer le temps de chargement du programme de l'unité de disque vers la mémoire, ce qui est particulièrement intéressant pour des gros programmes ou des programmes utilisés très fréquemment.
- 2 — Il est inutile que le programme soit présent sur les volumes disque utilisés, d'où une économie de place.
- 3 — Dans certains cas, le même programme peut être utilisé simultanément par plusieurs postes d'où une économie de place en mémoire centrale. C'est par exemple le cas avec l'exécuteur B.A.L.

Il existe trois niveaux de possibilités supportés par les programmes :

- Le programme ne peut pas être installé en mémoire centrale. Cette fonctionnalité lui est donc totalement interdite. C'est le cas de certains programmes utilitaires de PROLOGUE pour lesquels cette fonctionnalité n'est pas justifiée et de tous les programmes qui ne sont pas des programmes binaires PROLOGUE, c'est-à-dire dont le type n'est pas "X". C'est le cas par exemple de l'utilitaire "CPS-X".
- Le programme peut être installé en mémoire centrale de manière permanente, mais il ne peut être partagé. Dans ce cas, si deux utilisateurs souhaitent exécuter simultanément ce programme, le premier demandeur utilisera la copie résidente du programme mais le suivant utilisera implicitement la copie sur disque de ce programme, d'une façon transparente pour l'utilisateur. C'est le cas par exemple de l'utilitaire VPMI-X.

- Le programme peut être installé en mémoire centrale de manière permanente et peut être partagé. Dans ce cas, si plusieurs utilisateurs exécutent ce programme simultanément, c'est la même copie du programme en mémoire centrale qui sera utilisé. C'est le cas par exemple de l'exécuteur B.A.L. EX-X.

Le chapitre 4 indique pour chacun des utilitaires de PROLOGUE le niveau des possibilités qu'il supporte.

L'installation d'un programme en mémoire se fait par configuration du système de la façon suivante : il faut déclarer le programme dans le fichier "SYSCONF-S" à l'aide du mot-clé "INTR" :

"— Prog < INTR"

Il est nécessaire de faire une déclaration pour chacun des programmes à installer. ATTENTION : il faut s'assurer que le programme le permet effectivement, et donc qu'il est du type "X" et que la capacité mémoire de votre ordinateur vous le permet. L'utilitaire STMEM, décrit dans le chapitre 4, permet de connaître l'occupation d'un programme en mémoire.

Le chapitre 7 (Configuration et Installation de PROLOGUE) décrit de façon détaillée la structure du fichier "SYSCONF-S".

Après configuration du système conformément à vos besoins, l'utilisation d'un programme résident est absolument transparente pour l'utilisateur. Lorsque vous désignez un programme dans votre commande, l'interpréteur de commandes recherche automatiquement et en premier le programme en mémoire. Il n'y a donc pas de modification de la syntaxe d'appel des programmes.

Cependant, si dans la désignation du programme l'unité de disque est donnée explicitement, le programme sera forcément chargé à partir de cette unité de disque, même s'il est présent en mémoire centrale. Supposons par exemple que l'utilitaire CP soit installé en mémoire. La commande

"—> CP,param<rc>"

recherchera CP en mémoire, alors que la commande

"—> FLO.CP,param<rc>"

recherchera CP sur l'unité de disquettes "FLO".

3.3.5. Exécution de commandes par B.A.L. ou par la commande ASG.

Nous venons de voir que l'interpréteur de commandes permet d'exécuter les utilitaires de PROLOGUE ainsi que les programmes d'applications au moyen de commandes d'appel frappées au clavier.

Il existe deux autres façons d'exécuter des programmes sous PROLOGUE :

- Un programme B.A.L. peut lui-même charger et exécuter un utilitaire PROLOGUE sans avoir à revenir à l'interpréteur de commandes PROLOGUE. Ceci permet, tout en restant dans une même application, d'exécuter des fonctions de base du système : Recopie de fichiers, catalogue, etc. De plus la syntaxe d'appel de l'utilitaire peut être complètement masquée par l'application ce qui évite d'avoir à la connaître. Des contraintes propres au langage B.A.L. existent en ce domaine. Reportez-vous au Manuel de Référence B.A.L. pour gérer ces commandes.
- Lorsque vous souhaitez exécuter fréquemment une séquence de commandes, il vous est possible d'enregistrer cette séquence dans un fichier source à l'aide d'un éditeur de texte (EDV) et d'en provoquer l'exécution à l'aide de l'utilitaire ASG. (Ceci ne fonctionne pas sous B.A.L.).

Lorsque vous activez une séquence de commandes par ASG, tout se passe comme si le clavier de votre console était remplacé par le fichier source contenant la séquence des commandes. Du coup, l'interpréteur de commandes et tous les programmes exécutés reçoivent les données à partir du fichier et non plus du clavier. L'enchaînement prend fin lorsque la fin de fichier est détectée ou lorsqu'une erreur fatale survient. La mise en œuvre de l'utilitaire ASG est décrite au chapitre 4.

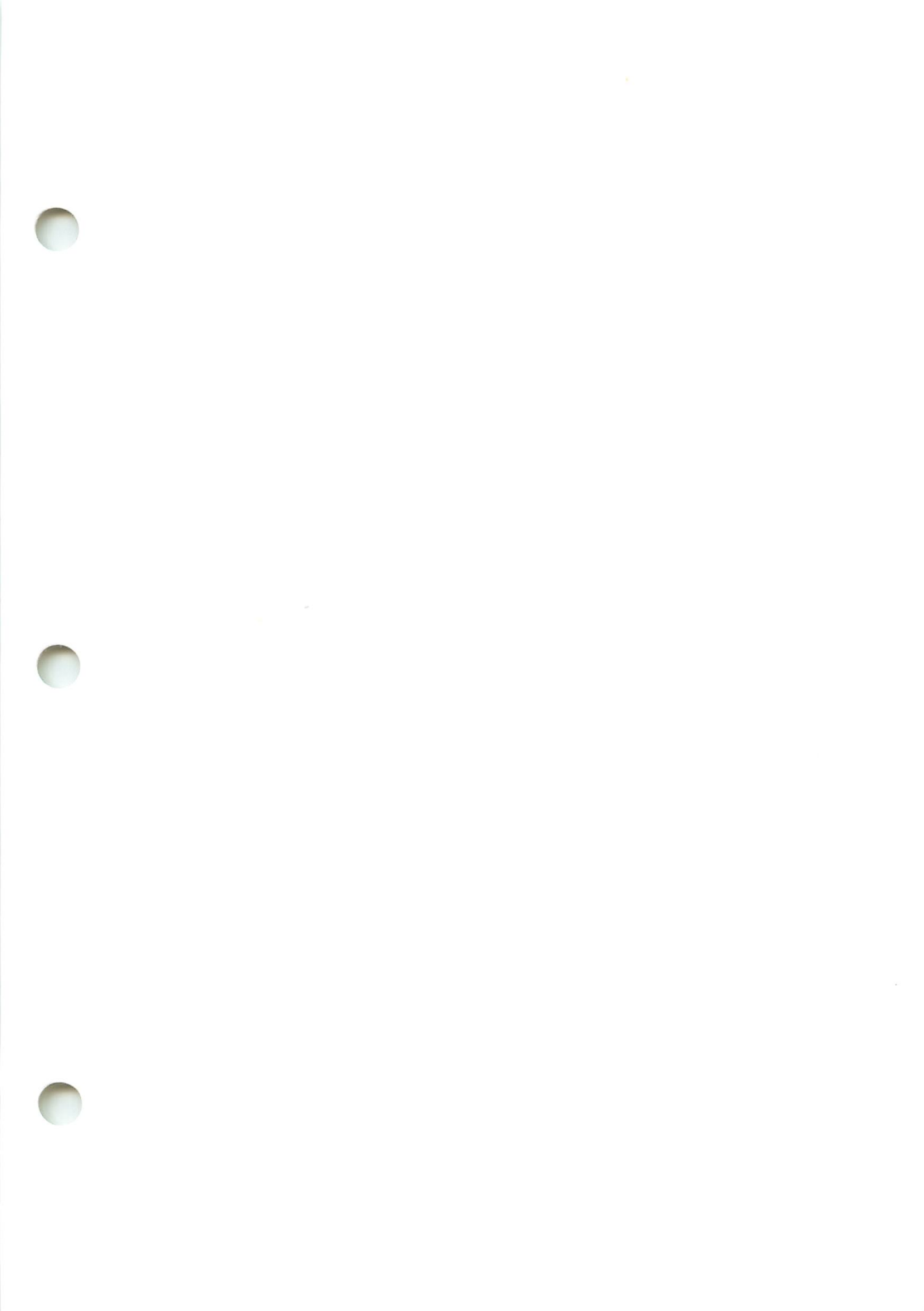
Le système PROLOGUE peut être configuré de façon à exécuter automatiquement une séquence préétablie de commandes sur chacun des postes lors du chargement du système. Il est donc possible d'activer automatiquement une application sans aucune intervention de l'utilisateur.

3.3.6. Les commandes et la sécurité

Les erreurs que vous pouvez commettre dans les commandes d'appel des programmes peuvent être dues soit à des erreurs de frappe, soit à une mauvaise désignation de fichier ou de volume ou une erreur de manipulation de volume. Par exemple, il arrive que les unités d'origine et destination soient interverties dans une commande de recopie, ou encore que les disquettes utilisées ne soient pas les bonnes.

Avant l'exécution effective d'une commande, PROLOGUE fait le maximum de contrôles sur la validité de la commande. Cependant il n'est pas possible de détecter les erreurs d'intention de l'utilisateur. En conséquence il est recommandé de suivre ces conseils :

- Protégez systématiquement contre les accès en écriture les volumes disquettes ou cartouches que vous n'utilisez qu'en lecture. Sur les disquettes ceci se fait en collant sur l'encoche de protection un adhésif.
- En cas de doute sur le contenu d'un volume, n'hésitez pas à utiliser la commande de catalogue "/" qui n'est jamais destructrice.
- Enfin procédez régulièrement à la sauvegarde de vos fichiers et de vos programmes. Ne travaillez jamais avec les originaux.



4.	LES COMMANDES DE BASE DU SYSTEME	4-1
4.1.	Généralités	4-1
4.2.	Les commandes courantes	4-4
4.2.1.	Catalogue de volume : "/"	4-4
4.2.1.1.	Catalogue complet de volume	4-5
4.2.1.2.	Catalogue partiel de volume	4-13
4.2.2.	Manipulation des fichiers et des volumes : "CP"	4-17
4.2.2.1.	Création d'un volume (CV)	4-19
4.2.2.2.	Modification du nom d'un volume (RV)	4-29
4.2.2.3.	Duplication d'un volume (DV)	4-34
4.2.2.4.	Duplication d'un ou plusieurs fichiers (DF)	4-44
4.2.2.5.	Création d'un fichier (CF)	4-54
4.2.2.6.	Suppression d'un ou plusieurs fichiers (SF)	4-58
4.2.2.7.	Modification du nom d'un fichier (RF)	4-64
4.2.3.	Visualisation et mise à jour de la date : "DATE"	4-69
4.2.4.	Visualisation et mise à jour des implicites : "VPMI"	4-73
4.2.5.	Copie Physique du Support : "CPS"	4-79
4.2.5.1.	Copie intégrale de support	4-79
4.2.5.2.	Copie partielle de support	4-82
4.2.6.	Sauvegarde des fichiers : "SV"	4-90
4.2.7.	Restauration des fichiers : "RT"	4-95
4.2.8.	Déconnexion du clavier au profit d'une commande : "ASG"	4-101
4.2.9.	Gestion des états à imprimer : "SPOOL"	4-111
4.2.9.1.	Activation du "SPOUL"	4-113
4.2.9.2.	Abandon d'une édition en cours	4-122
4.2.9.3.	Désactivation du "SPOUL"	4-125
4.2.9.4.	Insertion d'un fichier pour édition	4-129
4.2.9.5.	Utilisation du "SPOUL" en multiposte	4-133
4.2.10.	Emulation d'un volume en mémoire : "MEMDISK"	4-136

4.3.	Les commandes à usage système	4-137
4.3.1.	Visualisation et mise à jour des secteurs disque : "PATCH"	4-137
4.3.2.	Etat d'un support périphérique : "STATUS"	4-151
4.3.3.	Visualisation des partitions mémoire : "STMEM"	4-158

4. LES COMMANDES DE BASE DU SYSTEME

4.1. GENERALITES

Avec le système d'exploitation PROLOGUE, sont proposés des utilitaires, c'est-à-dire des programmes système, qui vous permettent d'effectuer des opérations de base telles que créer un volume, consulter le catalogue d'un volume, sauvegarder vos fichiers, créer des fichiers source, lancer des impressions différées,... Ces utilitaires peuvent s'exécuter au moyen de commandes lancées sous l'interpréteur de commandes, lorsque votre système est chargé et qu'apparaît le signe " $\rightarrow$ ".

Tous ces utilitaires sont des fichiers exécutables de suffixe "X" ou "T", ce dernier nécessitant la présence de l'exécuteur B.A.L. Certains de ces utilitaires sont résidents en mémoire parce que chargés par le configurateur à partir de l'unité implicite système. Pour les autres, la frappe de la commande provoquera leur chargement à partir de l'unité implicite programme de votre poste avant leur exécution ou d'une autre unité si celle-ci est spécifiée dans votre commande.

Nous pouvons distinguer deux types d'utilitaires :

- Ceux d'usage courant et dont la manipulation ne nécessite pas de connaissances informatiques approfondies. Ils feront l'objet du premier chapitre (4.2. les commandes courantes).
- Ceux d'usage système, d'approche plus délicate. Ils s'adressent à des utilisateurs ayant des notions système non pas à cause de leur mise en œuvre qui est aussi simple que pour les précédents, mais de par leur complexité d'interprétation. Ils feront l'objet du deuxième chapitre (4.3. les commandes à usage système).

* Rappel des conventions utilisées dans ce manuel

<TOUCHES> désigne une touche du clavier à activer
<rc> est la touche "Retour-chariot"
<ESC> est la touche "ESCAPE" ou "SORTIE"... etc.

ESCAPE ou ESC ou SORTIE désigne une touche dont la valeur hexadécimale est 1B (27 en décimal). Voyez les spécifications du constructeur pour votre machine.

-> représente la flèche PROLOGUE sollicitant une commande de la part de l'utilisateur.
- désigne la position courante du curseur (à ne pas confondre avec le tiret "-").
[] indique une option. Tout ce qui est compris entre ces deux crochets est facultatif.
{ } indique le choix entre plusieurs options, mais une d'entre elles est obligatoire.

* Entrée d'une commande sous l'interpréteur de commandes

Une commande se présente de la façon suivante :

-> CMD[paramètres et options]<rc>

où :

CMD représente le nom de la commande. C'est aussi le nom de l'utilitaire. Sa syntaxe est celle d'un descripteur de fichier avec le suffixe facultatif.

[paramètres et options] ils sont propres à chaque commande et indiquent des fonctionnalités différentes à exécuter. Ils sont séparés les uns des autres par des virgules.

<rc> c'est la touche de validation de la commande. Une fois frappée, la commande est analysée par l'interpréteur de commandes, puis exécutée ou refusée en cas d'erreur. Dans ce cas un message est retourné à l'utilisateur.

Toute commande peut être corrigée tant que la touche <rc> n'a pas été frappée. Au moyen des touches de déplacements de curseur, vous pouvez vous positionner sur le ou les caractères à modifier et corriger vos erreurs. Mais attention, au moment de frapper <rc>, votre curseur doit se trouver en fin de la chaîne de caractères représentant votre commande. Tous les caractères qui se trouvent à gauche du curseur seront pris en compte et feront donc partie intégrante de la commande, par contre tous les caractères qui se trouvent à droite du curseur, ainsi que le caractère pointé par celui-ci sont ignorés.

Une commande peut être frappée, indifféremment, en caractères majuscules ou minuscules. Il en va de même pour les paramètres, les options et les réponses de l'utilisateur aux messages des utilitaires. Les exceptions à cette règle seront signalées à l'utilisateur lors de la description de la commande.

4.2. LES COMMANDES COURANTES

4.2.1. Catalogue de volume : “/”

Présentation

L'utilitaire “/” (symbole de la barre oblique), vous permet de connaître la composition des volumes présents sur les supports disque au cours de votre exploitation.

Les différentes options associées à cet utilitaire vous permettent d'obtenir :

- Un catalogue complet du volume ou un catalogue partiel concernant un fichier ou un groupe de fichier.
- des informations relatives à la date de création ou mise à jour des fichiers ainsi que leur composition en blocs et granules.
- Un affichage des résultats à l'écran ou leur édition sur l'imprimante implicite de votre poste.

L'affichage de toutes ces informations, qui se présente en défilement continu sur votre terminal, peut être momentanément interrompu au moyen de la touche ESC (ou SORTIE). A partir de ce moment-là, l'utilisateur peut poursuivre l'affichage en actionnant une nouvelle fois cette touche ou abandonner et revenir à l'interpréteur de commandes en frappant la touche <rc>.

Sachez que “/” est une commande qui n'opère qu'en lecture. Elle peut donc être utilisée sans danger sur tout support, même protégé en écriture.

L'utilitaire “/” n'autorise pas une exécution différée de son chargement en mémoire.

L'utilitaire “/” peut être installé en mémoire de manière permanente lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable et il y a partage du programme entre les différents postes utilisateurs. Cette installation se fait par configuration du fichier SYSCONF-S. Reportez-vous dans le chapitre 7 au paragraphe 7.4.5. “Configuration des utilitaires résidents” pour plus de détails.

Nous allons maintenant détailler les différentes commandes d'appel de l'utilitaire "/". Deux types de commandes se présentent, décrites séparément :

- catalogue complet de volume,
- catalogue partiel de volume,

4.2.1.1. Catalogue complet de volume

Commande d'appel

—> [unité1.]/[.unité2][,options]<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire "/" est recherché afin d'être chargé en mémoire. Si "unité1" est omis, "/" sera recherché sur l'unité implicite programme de votre poste, ou en mémoire s'il est résident.

/ nom du fichier binaire correspondant ayant comme suffixe 'X'.

unité2 indique où se trouve le volume dont vous souhaitez connaître le catalogue. Si "unité2" est omis, le volume adressé sera celui de l'unité implicite fichier de votre poste.

options :

LIS=LO permet d'obtenir une impression des résultats sur l'imprimante implicite plutôt que sur l'écran. Par défaut l'affichage se fait sur l'écran.

GR permet d'obtenir, pour chaque fichier sélectionné par votre commande, la date de création ou de dernière mise à jour réalisée sur ce fichier ainsi que la liste des blocs composant ce fichier.

Ces deux options peuvent être employées dans un ordre quelconque dans la même commande.

Donnons quelques exemples :

Exemple 1 :

-> /<rc>

Cette commande vous permet d'afficher sur l'écran le catalogue complet du volume présent sur l'unité implicite fichier.

Après une exécution correcte de la commande, votre écran se présente par exemple comme suit :

-> /,FL0

Catalogue du volume Version 3.4b

Copyright (c) 1983 BULL Micral

Vol:PROLOG22 date:6.11.1984 Nbfic:63/20 Lg,Gr:16 Occup:152/95

/-X	ASG-X	CP-X	CPS-X	DATE-X
EDV-X	EX-X	FICSI-D	FICSI-I	PATCH-X
RTSI-X	SI-X	SPOOL-X	SVSI-X	SYSCONF-S
SYSCONF-X	SYSTEME	TFP-T	TSTSI-T	VPMI-X

-> .

Vol : indique le nom du volume, ici "PROLOG22"

date : indique la date de création du volume

Nbfic : indique le nombre maximum de fichiers que peut contenir ce volume, suivi du nombre de fichiers présents. Ici le volume peut contenir un maximum de 63 fichiers, il en contient effectivement 20.

Le nombre maximum de fichiers est défini lors de la création du volume par la commande "CP,CV,...". Le nombre maximum de fichiers ne peut excéder 895. La valeur par défaut est 63.

Lg,Gr : indique la taille des granules en nombre de secteurs. Les granules ont ici une taille de 16 secteurs (de 256 octets). La taille des granules est constante pour l'ensemble du volume et est définie lors de la création du volume par la commande "CP,CV,...". les valeurs possibles sont des puissances de 2 :

1,2,3,4,8,16,32,64,128,256.

La taille des granules ne peut pas être modifiée après la création du volume.

La valeur par défaut est 16.

Occup : indique le nombre total de granules composant le volume, suivi du nombre effectif de granules utilisés. Dans l'exemple ci-dessus, 95 granules ont été utilisés sur les 152 que comprend le volume. Lorsque le volume est plein, ces deux nombres sont évidemment égaux.

La liste de tous les fichiers présents sur le volume suit ces informations. Les fichiers sont présents dans l'ordre alphabétique des noms et des suffixes. Vous remarquerez que les clés d'accès ne sont évidemment pas visualisées.

Dans le cas où l'utilitaire "/" n'est pas présent sur l'unité implicite programme de votre poste, vous devez spécifier le nom de cette unité dans votre commande :

-> fl1./,fl0<rc>

Exemple 2 :

-> /,fl0,GR<rc>

Cette commande vous permet d'obtenir le catalogue complet du volume présent sur l'unité "fl0" qui n'est probablement pas l'unité implicite fichier. L'option "GR" vous permet de visualiser la date de dernière mise à jour et la composition en blocs et granules de chaque fichier.

Après une exécution correcte de la commande pour un volume identique à l'exemple précédent, vous obtenez les informations suivantes :

-> /,FL0,GR

Catalogue du volume Version 3.4.d

Copyright (c) 1983 BULL Micral

Vol: Prol/23a	date: 28.6.1985	Nbfic: 63/7	Lg.Gr: 16	Occup: 152 133
SYSTEME	: 26.6.1985	13 :	(16, 29)	
PROGR-ASM	: 10.6.1985	38 :	(40, 53) (112, 117) (120, 123) (128, 131)	
	:	:	(144, 145) (4, 5) (8, 9) (12, 13)	
	:	:	(36, 37)	
CP-X	: 21.6.1985	4 :	(72, 78)	
VPMI-X	: 12.2.1985	2 :	(88, 89)	
/-X	: 17.4.1985	2 :	(96, 97)	
FICSI-I	: 28.6.1985	69 :	(84, 87) (100, 103) (108, 111) (124, 127)	
	:	:	(132, 135) (140, 143) (148, 151) (2, 3)	
	:	:	(6, 7) (10, 11) (14, 15) (30, 31)	
	:	:	(34, 35) (38, 39) (54, 55) (58, 59)	
	:	:	(62, 63)	
	: Granule 78	:	(66, 67) (70, 71) (82, 93) (90, 91)	
	:	:	(94, 95) (98, 99) (118, 119) (138, 139)	
	:	:	(146, 147) (29, 29) (33, 33) (79, 79)	
FICSI-D	: 28.6.1985	2 :	(92, 93)	

->

La première colonne donne le nom et le suffixe du fichier. La colonne suivante donne soit :

- La date de dernière mise à jour opérée sur le fichier, lorsque cette information est sur la même ligne que le nom du fichier.
- Sinon c'est un numéro de granule qui est inscrit en indiquant que le fichier dépasse 18 blocs d'extensions.

La troisième colonne indique le nombre de granules composant le fichier, ce qui permet de déterminer son encombrement. Les colonnes suivantes donnent la composition du fichier en blocs et granules. Chaque couple de nombres entre parenthèses identifie un bloc. Le premier nombre est le numéro du premier granule du bloc, le suivant celui du dernier. Le nombre de blocs composant un fichier est variable. L'affichage de la liste des blocs peut se poursuivre sur plusieurs lignes si le nombre de blocs est important. Lorsque l'option "GR" est utilisée, les fichiers sont données dans l'ordre de leur mémorisation sur le volume.

Dans l'exemple ci-dessus, le fichier "FICSI—I" a été modifiée le 28 juin 1985. Il est constitué de 69 granules dont les 48 premiers sont décrits en 17 blocs dans le descripteur et les 21 autres sont décrits dans le granule 78 en 12 blocs.

Exemple 3 :

—>,FLO

Catalogue du volume Version 3.4b

Copyright (c) 1983 BULL Micral

COPIE

Vol: UTILIT1date: (1 = dernier)Nbfic: 895/10Lg.Gr: 32Occup: 76/21

SFGR-X	SGLOG-X	SI-X	SPL01-X	SPOOL-X
STATUS-X	STMEN-X	SV-X	SVSI-X	SYSCONF-X
—>				

Le volume se trouvant sur le support "FLO" est un volume de sauvegarde généré par la commande "SV" (voir paragraphe 4.2.6. de ce chapitre). "/" le signale en affichant "*COPIE*" et en indiquant le numéro de volume de la sauvegarde. Ici, c'est le premier volume et le dernier. "/" l'a signalé en affichant "(1=dernier)". Si la sauvegarde avait été faite sur plusieurs volumes seul le numéro de volume écrit entre parenthèses aurait été affiché.

Messages d'exécution

L'utilitaire "/" n'a pas de messages d'exécution propres autre que l'affichage des noms de fichiers accompagnés, si l'option "GR" est utilisée, de leurs emplacements respectifs sur le disque. Lorsque l'exécution de "/" s'est effectuée sans incident, le caractère "/" est affiché sur l'écran, ou sur l'imprimante si l'option "LIS=LO" a été spécifiée.

Messages d'erreurs

"/" est un utilitaire de consultation et de visualisation. Les messages d'erreurs sont donc limités aux messages système et à un message relatif à la disponibilité de l'imprimante.

* "Erreur de Commande n° 30 Syntaxe Incorrecte"

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

— deux unités spécifiées

Exemple :

—> /,MD0,FL0

Erreur de Commande n° 30 Syntaxe Incorrecte

— unité inexistante

Exemple :

—> /,FL2

Erreur de Commande n° 30 Syntaxe Incorrecte

Solution :

Corriger la syntaxe de votre commande, car l'unité de disque souple FL2 n'existe pas.

* "Imprimante non prête, sélectionnée (o/n) ?:"

Cause :

L'option "LIS=LO" a été utilisée dans la commande, mais aucune ligne n'a pu être écrite sur l'imprimante. L'impression a été temporairement arrêtée par un incident mécanique, ou une mise hors circuit de l'imprimante.

Exemple :

—> /,LIS=LO
Catalogue du volume Version 3.4b
Copyright (c) 1983 BULL Micral
imprimante non prête, sélectionnée (O/N) ?:

Solution :

Assurez-vous que votre imprimante est bien connectée et sélectionnée avant de frapper "O". Si ce n'est pas le cas, abandonnez la commande en frappant "N"; une erreur système numéro 1 sera alors affichée confirmant l'unité imprimante non prête.

Vérifiez que le papier est correctement placé dans l'imprimante.

* "Erreur fichier n° 54 Structure Volume Incorrecte"

Cause :

Le volume se trouvant sur le support désigné dans la commande n'est pas un volume PROLOGUE.

Exemple :

—> /,LIS=LO
Catalogue du volume Version 3.4b
Copyright (c) 1983 BULL Micral
Erreur Fichier n° 54 Structure Volume Incorrecte

Solution :

Créez votre volume (au moyen de la commande CP,CV,...)
ou monter le volume adéquat.

Remarques et conseils :

* Mise en œuvre de l'utilitaire “/”

Cet utilitaire est, avec l'utilitaire “CP”, celui qui est le plus fréquemment utilisé. A tout moment, il peut être utile de connaître le contenu des volumes utilisés. La ligne d'informations générales est importante car elle permet de suivre le taux de remplissage de votre volume et de se prémunir d'un risque de débordement.

Le débordement peut être un débordement de la capacité totale du volume c'est-à-dire lorsque le nombre de granules occupés est égal au nombre total de granules : “Occup:152/152”. Dans ce cas, il est nécessaire de créer et d'utiliser des volumes supplémentaires. Le débordement peut être également un débordement de la capacité du catalogue, c'est-à-dire lorsque le nombre de fichiers créés est égal au nombre maximum de fichiers :

“Nbfic: 63/63”.

Un tel débordement peut être évité par un dimensionnement correct de la capacité du catalogue, lors de la création du volume (commande CP,CV,...). Cette capacité ne peut être modifiée après la création du volume.

Note :

Certains utilitaires (tels EDV, TR, ... etc) créent des fichiers temporaires. Un débordement de volume peut alors apparaître alors que le nombre de fichiers n'est pas au maximum.

* Utilisation de l'option “GR”

Il ne vous est pas indispensable de connaître la composition des fichiers en blocs et granules. Cependant cette information met en évidence la dispersion du fichier sur l'espace disque. Un fichier comportant de nombreux blocs engendrera des temps d'accès plus longs. De plus si le fichier est amené à grossir, il y a risque que la limite des 18 blocs d'extension soit atteinte (si celle-ci a été choisie à la configuration), ce qui provoque une erreur fatale lors de l'exploitation du fichier.

L'usage d'une taille supérieure de granule permet de diminuer sensiblement ce risque. Enfin, l'utilitaire “REORG” décrit plus loin permet un recompactage global des fichiers sur le volume.

4.2.1.2. Catalogue partiel de volume

Commande d'appel

—> [unité1.]/,desfic[,options]<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire "/" est recherché afin d'être chargé en mémoire. Si "unité1" est omis, "/" est recherché sur l'unité implicite programme de votre poste.

/ nom du fichier binaire correspondant ayant comme suffixe "X".

desfic désigne un fichier unique ou un groupe de fichiers que vous souhaitez sélectionner par cette commande. La désignation d'un groupe de fichiers se fait par l'usage de caractères de substitution "*" ou "?" (voir chapitre 2, le paragraphe 2.2.2.3. "Les caractères de substitution").

Les options de cette commande sont identiques aux options de la commande précédente : catalogue complet.

Exemple 1 :

—> /,FL0.S*-X,GR<rc>

Cette commande sélectionne l'ensemble des fichiers de l'unité "FL0" dont le nom commence par la lettre "S" et dont le suffixe est "X".

On obtiendra par exemple les informations suivantes :

Vol: TESTUTIL	date: 7.11.1984	Nbfic: 63/15	Lg.Gr: 16	Occup: 152/58
SPOOL-X	: 9. 7.1984	2: (64, 65)		
STATUS-X	: 9. 7.1984	2: (72, 73)		
SV-X	: 10. 9.84	4: (80, 83)		
STMEN-X	: 13. 7.1984	2: (88, 89)		
SYSCONF-X	: 5.11.1984	2: (96, 97)		
SVSI-X	: 16. 7.1984	4: (120, 123)		
SI-X	: 17. 9.1984	4: (128, 131)		

—> .

Exemple 2 :

—> /,fl0.*optics*.LIS=LO<rc>

Cette commande permet d'obtenir une édition sur l'imprimante implicite de votre poste du catalogue des fichiers dont le nom se termine par "optics" et dont le suffixe est quelconque.

L'impression obtenue sera par exemple la suivante :

Vol: prologue	Nbfic: 63/30	Lg.Gr: 16	Occup: 152/118
OPTICS-D	OPTICS-I	OPTICS-R	OPTICS-V
ROPTICS-I			ROPTICS-D

Exemple 3 :

—> /,FL0.SP?O??.GR

Cette commande, utilisant les caractères de substitution (voir paragraphe 2.2.2.3. au chapitre 2), permet d'obtenir tous les fichiers avec les restrictions suivantes :

- Le nom est de 5 caractères maximum.
- Le nom commence par "SP".
- Le quatrième caractère du nom est un "O".
- Le suffixe est d'un caractère maximum.

Nous aurons donc, par exemple :

Vol: TESTUTIL	date: 7.11.1984	Nbfic: 63/16	Lg.Gr: 16	Occup: 152/59
SPOOL-X	: 9. 7.1984			2: (64, 65)
SPOOL1-X	:			2: (136, 137)
SPTO	: 7.11.1984			1: (144, 144)

—> —

Messages d'exécution

Voir paragraphe précédent "Catalogue complet d'un volume".

Messages d'erreur

Ces messages sont identiques à ceux décrits dans le paragraphe précédent "Catalogue complet d'un volume".
Voici quelques exemples pour illustrer ceux-ci.

Exemple 1 :

—> /,*S*-X

Erreur de commande n° 30 Syntaxe Incorrecte

—> —

L'utilisateur a utilisé deux fois le caractère de substitution "*" dans le nom de fichier. Il faut utiliser "*" au maximum une fois dans le nom et une fois maximum dans le suffixe.

Exemple 2 :

—> /,*.?

Erreur de commande n° 30 Syntaxe Incorrecte

On ne peut utiliser dans une même commande les deux caractères de substitution "*" et "?".

Exemple 3 :

—> /,a* z

Erreur de commande n° 30 Syntaxe Incorrecte

—> —

Le caractère de substitution "*" doit être employé pour désigner le début ou la fin du nom du fichier, le début ou la fin du suffixe de fichier.

Remarques et conseils :

* Catalogue d'un fichier unique

Cette commande permet de connaître la date de dernière mise à jour du fichier ainsi que sa composition en blocs. De plus elle permet de vérifier très simplement l'existence effective d'un fichier sur un volume.

Exemple :

—> /,FL0.PROLUTI-IMP

Cette commande vous donne les caractéristiques du fichier “PROLUTI-IMP” localisé sur l’unité “FL0” sous la forme suivante :

Vol: UTILIT1 date: 4.11.1984 Nbfic 895/197 Lg.Gr: 32 Occup: 1374/625
PROLUTI-IMP :7.11.1984 36:(1176,1183)(1183,1191)(1196,1199)(1204,1207)
(1212,1215)(1220,1223)(1228,1231)(1236,1239)

—>

La première colonne donne le nom du fichier. La suivante indique la date de dernière mise à jour. La troisième colonne donne la taille du fichier en nombre de granules.

Les colonnes suivantes décrivent la composition du fichier en blocs et granules.

Ces informations sont identiques à celles fournies par la commande de catalogue global suivi de l’option “GR”. Vous remarquerez que dans le cas d’une commande “/” sur un fichier unique, l’option “GR” est implicite.

4.2.2. La manipulation des fichiers et des volumes : "CP"

Présentation

L'utilitaire CP vous permet de manipuler les fichiers PROLOGUE créés et utilisés par vos applications. Il vous permet de travailler soit au niveau d'un fichier ou d'un groupe de fichiers, soit au niveau global d'un volume. CP vous permet par exemple de dupliquer, supprimer un fichier, créer ou renommer un volume.

Le choix de la fonction que vous souhaitez exécuter par CP se fait au moyen d'un mot-clé que vous devez placer dans la commande d'appel de CP, immédiatement derrière son nom.

Les fonctions et mots-clés possibles sont les suivants :

— Création d'un volume	CV
— Modification du nom d'un volume	RV
— Duplication d'un volume	DV
— Création d'un fichier	CF
— Modification du nom d'un fichier	RF
— Suppression d'un fichier ou d'un groupe de fichiers	SF
— Duplication d'un fichier ou d'un groupe de fichiers	DF

Les trois premières fonctions portent sur des volumes, les suivantes portent sur des fichiers ou des groupes de fichiers.

Utilisation répétitive de CP

Quelque soit la commande choisie, sa mise en œuvre peut être faite en deux temps :

- 1) Appel en mémoire de l'utilitaire
—> CP <rc>

Lorsque le chargement est terminé, une virgule apparaît derrière CP, indiquant que PROLOGUE attend la suite de la commande.

- 2) Suite de la commande et exécution
—> CP, paramètres ...<rc>

Cette procédure vous permet, éventuellement de changer le volume sur lequel se trouve l'utilitaire CP lorsque celui-ci a été chargé. Ceci peut être utile lorsqu'on souhaite faire, par exemple, une duplication de fichiers et que CP-X ne se trouve ni sur le volume d'origine, ni sur le volume de destination. De plus, après que la commande ait été exécutée, l'utilitaire CP reste chargé en mémoire en attente d'une nouvelle commande.

- > CP,paramètre<rc> -
- > CP,paramètre2<rc> -
- > CP,-

La sortie de ce mode répétitif se fait par la frappe immédiate d'un retour chariot <rc>, de la même façon que pour obtenir le numéro de version de CP.

Si vous souhaitez connaître le numéro de version de l'utilitaire CP que vous utilisez, procédez comme suit :

- Appelez l'utilitaire en mémoire sans paramètres
 - > CP<rc>
- Lorsque la virgule, indiquant la fin de chargement de l'utilitaire, est affichée, appuyez de nouveau sur le retour chariot, et vous obtiendrez ce qui suit :
 - > CP,<rc>
 - Manipulation de fichiers et de volumes Version x.x
 - Copyright (c) 1984 BULL Micral
 - > .

x.x est le numéro courant de version de l'utilitaire CP.

L'utilitaire CP peut être installé de manière permanente en mémoire lors de l'initialisation du système. Dans ce cas son exécution se fait sans chargement préalable du fichier correspondant et il y a partage du programme entre les différents postes utilisateurs. Cette installation se fait par la déclaration suivante dans le fichier source SYSCONF-S :

- CP < INTR

Pour plus de détails reportez-vous au paragraphe 7.4.5. "Configuration des utilitaires résidents" du chapitre 7.

4.2.2.1. Création d'un volume (CV)

Présentation

Cette fonction permet la création d'un volume PROLOGUE sur un support périphérique, opération nécessaire avant tout usage de ce support. C'est à cette occasion que vous indiquez les caractéristiques du volume (taille des granules et nombre maximum de fichiers).

Si la création de volume est faite sur un support déjà exploité, il y a perte de TOUS les fichiers du volume précédent.

L'usage de cette fonction peut être interdit par la configuration de votre système PROLOGUE. Un message vous signale cette interdiction et il y a abandon de la fonction.

A l'issue d'une déclaration de volume, le catalogue du volume est vide.

Commande d'appel

—> [unité1.]CP,CV,unité2,nomvol[,options]

où :

unité1	indique l'unité sur laquelle l'utilitaire CP est recherché afin d'être chargé en mémoire. Si "unité1" est omis, l'utilitaire CP est recherché sur l'unité implicite programme de votre poste.
CP	nom du fichier correspondant ayant comme suffixe "X".
CV	mot-clé spécifiant la fonction de création de volume.
unité2	indique l'unité sur laquelle le volume doit être créé.
nomvol	est le nom du volume. Il comporte 1 à 8 caractères alphanumériques, ainsi que les caractères / et \$.

Options :

LGR=Lg permet de définir la taille des granules sur l'ensemble du volume. Cette taille est donnée en nombre de secteurs PROLOGUE, et peut prendre les valeurs décimales suivantes : 1, 2, 4, 8, 16, 32, 64, 128, 256 (puissances de 2). Si cette option est omise, la taille implicite des granules est de 16 secteurs.

PROLOGUE pouvant gérer jusqu'à 8176 granules sur un même volume, il faut que le produit $8176 \times \text{LGR}$ soit $\geq$ au volume à gérer.

NBFIC=nb permet de définir le nombre maximum de fichiers pouvant être créés sur le volume (plus on déclare un nombre de fichiers maximum grand, plus le catalogue prend de place sur le volume).

Le nombre maximum de fichiers ne peut excéder 895.

Les fichiers sont groupés par 32 dans le catalogue, sauf pour le premier groupe qui n'en contient que 31.

Le nombre maximum de fichiers est, de fait, un multiple de 32 moins 1 c'est-à-dire 31, 63, 95 etc...

Toutefois, une valeur quelconque est acceptée pour ce paramètre, l'utilitaire CP se chargeant d'ajuster cette valeur. Si cette option est omise, la longueur du catalogue est de 63 fichiers.

TYPE=9F permet de créer un volume dont l'indicatif sera 9F, c'est-à-dire qu'il ne pourra être composé que de fichiers de type 7-1, limité à 18 blocs d'extension.

Si cette option est omise, c'est un volume avec un indicatif A0 qui sera créé, c'est-à-dire un volume autorisant des fichiers de type 8-3 avec une date de dernière mise à jour.

PM permet le prémarquage du support sur lequel le volume doit être créé.

Cette option est indispensable si le support utilisé est neuf ou s'il a été prémarqué dans un format incompatible avec PROLOGUE, ou si l'on change de matériel (ex : disquette 600 K utilisée sur le lecteur 300 Ko).

Exemple 1 :

—> CP,CV,fl1,compta<rc>

Cette commande qui est parmi les plus courantes, crée un volume nommé “compta”, localisé sur l’unité périphérique FL1. Les options étant toutes omises, les caractéristiques du volume seront les suivantes :

Longueur des granules égale à 16 secteurs

nombre maximum de fichiers égal à 63.

Indicatif du volume A0. (Nom de fichier 8—3 autorisé, date gérée, nombre de blocs d’extension illimité).

Il n’y a pas de prémarquage préalable du support. L’utilitaire CP est recherché sur l’unité implicite programme de votre poste.

Si vous examinez alors le catalogue du volume créé, au moyen de l’utilitaire /, il apparaîtra comme suit :

—> /,fl1 Version : 3.2.

Vol: compta date: 25.11.1984 Nbfic: 63/0 Lg.Gr:16 Occup: 152/2

63 est le nombre maximum de fichiers (taille du catalogue).

0 est le nombre de fichiers présents sur le volume.

16 indique la longueur des granules sur ce volume.

152 indique le nombre total de granules disponibles sur ce volume.

Cette valeur multipliée par la taille du granule donne la capacité totale du support qui est de 608 K dans cet exemple.

2 est le nombre de granules occupés par le catalogue lui-même, en l’absence de tout fichier.

Exemple 2 :

—> md0.CP,CV,fl0,VOL/01,TYPE=9F,PM,NBFIC=16<rc>

Dans ce cas l'utilitaire CP est recherché sur l'unité MD0, qui probablement n'est pas votre unité implicite programme car il serait alors inutile de l'indiquer. Le volume "VOL/01" est créé sur l'unité FL0 avec un indicatif à 9F. Il y a prémarquage préalable du support, la taille du catalogue est de 16 fichiers.

La visualisation de ce volume donne :

—> md0./,fl0 Version : 3.2.

Vol: VOL/01 date 25.11.1984 Nbfic: 31/0 Lg:Gr: 16 Occup: 152/2

Vous remarquerez que le nombre de fichiers demandé était de 16 et que la longueur effective du catalogue est de 31, ceci est dû au fait que les fichiers sont groupés par 32 dans le catalogue. Que ce soit pour 16 fichiers ou 31 fichiers, la place occupée par le catalogue est identique.

Messages d'exécution

Lors de l'exécution d'une commande de création de volume, un certain nombre de messages sont affichés à l'écran par CP, vous informant de la progression de l'exécution ou vous invitant à valider votre choix. Lorsque l'exécution de la fonction s'est déroulée sans erreur, le symbole "—" est affiché à la suite de la commande et le contrôle est rendu à l'interpréteur de commandes ou à CP si vous êtes en mode répétitif.

Votre écran se présentera comme suit :

—> CP,CV,fl1,prolog —

—> .

- * Destruction de volume : si le support présent dans l'unité périphérique désignée dans la commande comporte déjà un volume PROLOGUE, le message suivant apparaît :

Détruire le volume nomvol (O-N) :

où "nomvol" est le nom du volume présent.

Si vous répondez par OUI (O<rc> ou o<rc>), le volume "nomvol" ainsi que TOUS ses fichiers sont détruits et remplacés par le volume que vous souhaitez créer. Toute autre réponse provoquera l'abandon de la commande sans destruction de l'ancien volume.

Avant de répondre OUI, assurez-vous plutôt deux fois qu'une, qu'aucun fichier du volume que vous allez détruire n'a d'importance !

Dans le cas où le volume présent est un volume de type COPIE (voir utilitaire SV), le message de validation devient :

Détruire volume **COPIE** nomvol num : x (O-N) : —

où x est le n° d'ordre du volume copié. Si ce volume est le dernier de la série, la mention "dernier" apparaît.

Prémarquage (O-N) : —

Si vous répondez OUI (O <rc> ou o <rc>), le message suivant s'affiche :

Sélectionnez le type de disquettes à formater (1-300K, 2-1,2 Mégas) : Entrez votre choix.

La commande est exécutée avec prémarquage préalable du support. Toute autre réponse provoquera l'abandon de la commande, laissant intact le support adressé.

Lorsque cette option est présente, le support adressé est supposé vierge donc illisible. Il n'y a donc pas de vérification de la présence éventuelle d'un volume PROLOGUE sur ce support. Le message "Détruire nomvol" ne peut donc apparaître. Compte tenu de l'effet irrémédiable d'un prémarquage, la validation de ce message doit être faite avec soin.

Messages d'erreur

Tant que l'opération de création de volume n'est pas terminée, une erreur fatale peut éventuellement survenir provoquant l'affichage d'un message d'erreur et l'abandon de la commande. Dans ce cas, quel que soit l'état d'avancement, le travail accompli n'est pas significatif et la création du volume doit être relancée après suppression des causes d'erreurs possibles.

Si l'erreur est une erreur système, non propre à CP, le libellé et le numéro d'erreur sont affichés et vous vous reporterez à l'annexe B afin de connaître sa signification et les solutions envisageables.

Exemple :

—> CP,CV,fl0,toto,NBFIC=31

Erreur périphérique n° 5 : protection écriture

—> -

Tentative d'écriture sur une disquette protégée.

Solution :

enlevez la languette de protection présente sur la disquette, après avoir vérifié son label.

Si l'erreur est une erreur propre à la fonction CV de l'utilitaire CP, un message d'erreur particulier est affiché. Nous allons donner ci-dessous la signification de ces messages et les solutions possibles.

* "Erreur de commande n° 30 syntaxe incorrecte"

Cause .

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

— mot-clé de fonction inexistant.

Ex : SV.

— nom d'unité incorrect.

Ex : fl123 ou DR1 ; en supposant que DR1 n'est pas supporté par PROLOGUE.

- nom de volume trop long.
Ex : ,MONVOLUME, ; 9 caractères
- caractères incorrects dans le nom de volume.
Ex : ,VOL-001, ; “-” interdit
- option incorrecte.
Ex : ,LGRA=32, ; LGRA : mot-clé inconnu.

Solution :
Corrigez la syntaxe de votre commande.

* “Fonction interdite”

Cause :

La fonction création de volume ne vous est pas accessible avec ce système PROLOGUE.

Solution :

Pas de solution possible si cette interdiction est voulue pour des raisons de sécurité. Sinon reconfigurer votre système ou utilisez un système adéquat (voir chapitre 7).

* “Prémarquage impossible sur cette machine”

Cause :

L'option PM n'est pas supportée par le contrôleur de la ressource périphérique adressée. Dans ce cas le prémarquage doit être fait préalablement avec un programme particulier à cet usage. Consulter la documentation propre à votre micro-ordinateur.

* “Longueur granule erronée”

Cause :

La taille du granule, choisie par le paramètre optionnel LGR= de votre commande n'est pas égale à l'une des valeurs imposées par PROLOGUE.

Exemple : —> CP,CV,md0,abcdef,LGR=48
(48 n'est pas une puissance de 2)

Solution :

Reportez-vous au paragraphe 2.1.2.2. “Organisation Logique” du chapitre 2 relatif à l'organisation des fichiers, puis relancez votre commande avec la taille de granule adéquat.

* “Granule trop petit”

Cause :

La taille des granules, définie par votre commande d'appel, soit par défaut soit explicitement à l'aide du paramètre LGR=, est trop petite pour permettre à PROLOGUE d'atteindre la capacité totale du support. En effet le nombre maximum de granules d'un volume ne pouvant excéder 8176, la taille du granule multipliée par 8176 doit être au moins égale à la capacité totale du support.

Exemple : Une taille de granule de 16 secteurs de 256 octets permettra de couvrir un support n'excédant pas 32 Méga-octets.

Solution :

Relancer l'opération de création de volume en choisissant une taille de granule plus importante.

* “Nb. erroné de fichiers”

•Cause :

Le nombre de fichiers, défini par le paramètre optionnel NBFIC= de votre commande est nul ou supérieur à 895.

Solution :

Relancer l'opération de création de volume avec une valeur appropriée de longueur de catalogue.

Remarques et conseils

* Utilisation de l'option "NBFIC="

Ce paramètre définit la taille du catalogue du volume qui ne peut être modifiée par la suite. Cette taille doit être déterminée en fonction de l'usage prévu du volume et de sa capacité.

Les fichiers étant groupés par 32 dans le catalogue (moins 1 pour le premier groupe), il est recommandé de donner des valeurs égales à un multiple de 32 moins 1, c'est-à-dire 31, 63, 95...

Pour fixer les idées, une capacité de 63 fichiers est une valeur raisonnable sur une disquette de 600 koctets mais probablement insuffisante sur un mini disque de 10 Moctets.

* Utilisation de l'option "LGR="

La taille des granules, définie par cette option, est constante sur l'ensemble du volume et ne peut être modifiée après la création du volume.

Sachant qu'un fichier occupe toujours un nombre entier de granules, le dernier granule alloué au fichier risque de ne pas être complètement rempli (place perdue). En conséquence, plus la taille des granules est petite, plus le risque évoqué précédemment est minime. Par contre, plus le granule est grand, moins il y a de risque qu'un fichier tombe en erreur de débordement des blocs d'extension lorsqu'il grandit. Il convient donc de choisir judicieusement la taille des granules. Toutefois, et si cela est en harmonie avec vos contraintes d'exploitation, il est intéressant de choisir une taille de granules égale à la taille de la piste physique du support utilisé. Consultez, pour cela la documentation propre à votre micro-ordinateur.

Exemple : LGR = 32

Tout fichier sur le support sera composé de $n \times 32$ secteurs. Il risque donc d'y avoir au pire 31 secteurs et 255 octets de place perdue par fichier. Par contre, vous êtes assurés d'avoir des fichiers dont la taille maximum sera dans le pire des cas 18×32 secteurs. Si vous voulez garantir une taille supérieure, il vous faudra augmenter cette taille de granules. Si par contre, vous êtes certains que le plus grand de vos fichiers sur la machine n'atteindra pas cette taille, réduisez la taille du granule.

* Utilisation de l'option "TYPE=9F"

Par défaut, les volumes sont donc créés avec un indicatif à A0. Ce qui permet d'avoir des fichiers au format 8-3, ainsi qu'une date de dernière mise à jour associée.

L'indicatif 9F mis en place par cette option limite les noms de fichiers au format 7-1 sans date de mise à jour. L'intérêt de ce format est que le volume peut être relu par des systèmes PROLOGUE 8 bits (à base de 8080 ou compatibles).

* Utilisation de l'option "PM"

Le prémarquage ne se fait en général qu'une fois, avant la première utilisation du support. Cependant le prémarquage d'un support permet de vérifier la qualité physique de ce support et de détecter d'éventuels secteurs invalides, pour les supports qui en assurent la gestion.

N'oubliez pas qu'un prémarquage est une opération qui peut durer de 1 à 10 minutes et que le support prémarqué est totalement effacé !

4.2.2.2. Modification d'un nom de volume (RV)

Présentation

Cette fonction vous permet de modifier le nom d'un volume PROLOGUE, après sa création par la fonction CV. L'état en cours du volume et des fichiers présents n'est pas modifié.

Dans le cas où la capacité théorique de la ressource adressée est supérieure à la capacité effective du volume, la fonction RV vous permet optionnellement d'ajuster la capacité du volume à sa valeur théorique. Ce cas de figure peu fréquent, sera discuté ci-dessous. L'usage de la fonction RV peut être interdit par la configuration de votre système PROLOGUE. Un message vous signale cette interdiction et il y a abandon de la fonction.

Commande d'appel

—> [unité1.]CP,RV,unité2,nomvol<rc>

où :

- unité1 indique l'unité sur laquelle l'utilitaire CP est recherché afin d'être chargé en mémoire. Si "unité1" est omis, l'utilitaire CP est recherché sur l'unité implicite programme de votre poste.
- CP est le nom du fichier correspondant ayant comme suffixe "X".
- RV est le mot-clé de la fonction modification du nom du volume.
- unité2 indique l'unité sur laquelle est présent le volume concerné.
- nomvol est le nouveau nom attribué au volume, en remplacement du nom précédent.

Exemple :

Supposons qu'un volume, dont le catalogue apparaît comme suit, soit présent sur l'unité FL0 :

Vol: ancnom	date: 25.11.84	Nbfic: 63/7	Lg.Gr: 16	Occup: 152/63
/-X	CP-X	CLIENTS-D	CLIENTS-I	ED-X
EX-X	PAIE-T			

"ancnom" est le nom actuel du volume.

La commande suivante :

—> CP,RV,fl0,nouvnom<rc>

donne un catalogue qui s'affichera comme suit, après une commande “/”.

Vol: nouvnom date: 25.11.84 Nbfic: 63/7 Lg.Gr: 16 Occup: 152/63

/-X	CP-X	CLIENTS-D	CLIENTS-I	ED-X
EX-X	PAIE-T			

“nouvnom” est maintenant le nom du volume. Toutes les autres caractéristiques sont inchangées.

Message d'exécution

Lors de l'exécution d'une commande de modification du nom d'un volume, un certain nombre de messages sont affichés à l'écran par CP, vous informant de la progression de l'exécution et vous invitant parfois à valider un choix. Lorsque la fonction s'est exécutée sans erreur, le symbole “—” est affiché à la suite de votre commande et le contrôle est rendu à l'interpréteur de commandes ou à CP si vous êtes en mode répétitif.

Votre écran se présente par exemple comme suit :

—> CP,RV,md0,DISFIX —

—> —

* Nom du volume :

Le nom du volume adressé est affiché avec validation de l'opération :

Renommer le volume : nomvol (O-N) :

Si vous répondez par OUI (O<rc> ou o<rc>), la modification du nom est effectuée. Tout autre réponse provoque l'abandon de la fonction sans modification du nom du volume.

* Ajustement de la capacité du volume :

Si la capacité de l'unité périphérique adressée, définie par PROLOGUE, s'avère être supérieure à la capacité effective du volume (nombre total de granules multiplié par la taille du granule), le message suivant est affiché :

Réajustement de nb. de granules sur le volume (O-N) : -

Si vous répondez par OUI (O<rc> ou o<rc>), le nombre total de granules est augmenté de façon à couvrir exactement la capacité du support. Tout autre réponse laisse intact le nombre de granules.

Message d'erreur

Dans le cas où une erreur fatale survient en cours d'exécution, un message d'erreur est affiché indiquant la nature de l'erreur et la commande en cours est abandonnée. Dans ce cas le travail accompli n'est pas significatif et la commande doit être relancée après avoir supprimé les causes d'erreurs possibles.

Si l'erreur est une erreur système non spécifique à CP, le libellé et le n° d'erreur sont affichés et vous vous reporterez à l'annexe B, afin de connaître sa signification et les solutions possibles.

Exemple :

-> CP,RV,f10,nouveau

Erreur fichier 54 : structure volume incorrecte

-> -

L'unité adressée par votre commande ne contient pas un volume PROLOGUE.

Si l'erreur est une erreur propre à la fonction RV de l'utilitaire CP, un message d'erreur particulier est affiché. Nous allons donner ci-dessous la signification de ces messages et les solutions possibles.

* "Erreur de commande n° 30 : syntaxe incorrecte"

Cause :

La syntaxe de votre commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction incorrect.
Ex : —> CP,SV,FL0,NOUVNOM<rc>
"SV" n'est pas une fonction de CP.
- nom d'unité incorrect.
Ex : —> CP,RV,FICTEST,VOLNOM<rc>
"FICTEST" n'est pas un nom valide d'unité périphérique.
- absence nom de volume.
Ex : —> CP,RV,md0<rc>
- nom de volume trop long.
Ex : —> CP,RV,fl0,ungrandnom<rc>
La longueur de "ungrandnom" est supérieure à 8 caractères.
- caractère incorrect dans le nom de volume.
Ex : —> CP,RV,fl0,vol-001<rc>
Le caractère "—" est interdit.

* "Fonction interdite"

Cause :

La fonction modifications du nom d'un volume ne vous est pas accessible avec ce système PROLOGUE.

Solution :

Pas de solution possible si cette interdiction est voulue pour des raisons de sécurité. Sinon reconfigurer votre système (voir chapitre 7) ou utiliser un système adéquat.

* "Erreur de fichier n° 54 : Absence de volume PROLOGUE"

Cause :

L'unité adressée par votre commande ne contient pas un volume PROLOGUE.

Solution :

Vérifiez la validité du support présent sur l'unité périphérique que vous adressez, et procédez éventuellement à une fonction de création de volume (CV).

* “Lg. granule erronée”

Cause :

La taille des granules, définie sur le volume adressé a une valeur incorrecte. Ceci est probablement dû à une détérioration du support rendant le volume illisible.

Solution :

Faites immédiatement une duplication de votre volume (commande CP, DV, ou CPS) si vous souhaitez récupérer des fichiers présents sur le volume, puis abandonnez ou prémarquez le support utilisé suivant sa nature.

* “Incompatibil. entre le nb. de granules du volume et le contrôleur qui le gère”.

Cause :

Le nombre total de granules déclaré sur le support adressé est supérieur au nombre total de granules admis par le système PROLOGUE gérant l'unité périphérique adressée, de manière incompatible avec le système PROLOGUE actuel.

Solution :

Dupliquez les fichiers que vous souhaitez conserver par la commande CP,DF. Toutefois vous ne pourrez dupliquer les fichiers alloués sur des granules au-delà de la limite actuelle du volume. Une tentative de duplication provoquera l'erreur :

“Erreur périphérique n° 4 : accès illégal”

Puis procédez à une création de volume sur l'unité adressée, rendant ainsi la taille du volume conforme au système PROLOGUE utilisé.

4.2.2.3. Duplication d'un volume (DV)

Présentation

Cette fonction permet la recopie de tous les fichiers d'un volume vers un autre volume sans modification des caractéristiques du volume destination qui peuvent être différentes (ex : taille des granules). Chaque fichier présent sur le volume origine est recopié successivement sur le volume destination en conservant ses caractéristiques (nom, suffixe, clés d'accès, nature). Si un fichier de même nom et de même suffixe est présent sur le volume destination, il est supprimé et remplacé par le fichier origine, à condition toutefois que ses clés d'accès soient identiques à celles du fichier origine ou inexistantes.

Si en cours de copie, le volume destination est saturé, il y a suppression du fichier en cours de copie et vous avez alors la possibilité de poursuivre la copie sur un nouveau volume que vous aurez préalablement créé et après avoir fait le changement de support nécessaire.

Un paramètre optionnel de la commande (mot-clé "RZ") vous permet de supprimer préalablement tous les fichiers qui se trouvent sur le volume de destination.

L'usage de la fonction de duplication de volume peut vous être interdit par configuration de votre système PROLOGUE. Un message vous signale cette interdiction et il y a abandon de la commande.

Commande d'appel

—> [unité1.]CP,DV,unité2,[DEST=]unité3[,option]<rc>

où :

- | | |
|--------|--|
| unité1 | indique l'unité sur laquelle l'utilitaire CP est recherché. Si "unité1" est absent, CP est recherché sur l'unité implicite programme de votre poste. |
| CP | nom du fichier binaire correspondant ayant comme suffixe "X". |
| DV | mot-clé spécifiant la fonction de duplication de volume. |
| unité2 | indique l'unité supportant le volume origine de la duplication. |
| DEST= | mot-clé optionnel précisant la nature de "unité3". |
| unité3 | indique l'unité supportant le volume destination de la copie. "unité3" doit être nécessairement différent de "unité2". |

option :

RZ supprime tous les fichiers présents sur le volume de destination.

Le nombre total de granules du volume destination est lié à la capacité du support utilisé et peut donc être différent du nombre total de granules du volume d'origine.

Si l'option "RZ" est absente, à l'issue de la duplication, le volume destination contiendra l'ensemble des fichiers du volume origine et des fichiers du volume destination.

Si l'option "RZ" est utilisée, à l'issue de la duplication, le volume destination contiendra uniquement l'ensemble des fichiers du volume origine, les fichiers du volume destination ayant été préalablement supprimés.

Exemple :

Vous souhaitez dupliquer les fichiers du volume "BASE001" sur le volume "BASESAV".

Le volume "BASE001" présent sur l'unité "FL0" contient les fichiers suivants :

VOL: BASE001 date: 25.11.84 Nbfic: 63/7 Lg.Gr: 8 Occup: 304/17
EXEMPLE-S EXEMPLE-T OUVREF-A OUVREF-S REVBIM-S
REVMEN-S REVTRI-S

Le volume "BASESAV", présent sur l'unité "FL1" contient les fichiers suivants :

Vol: basesav date: 25.11.84 Nbfic: 63/3 Lg.Gr: 4 Occup: 608/22
PASS01-S PASS02-S TEXT-S

Vous exécutez la commande de duplication suivante :

-> CP,DV,f10,f11<rc>

A l'issue de cette opération de duplication, et si l'exécution est correcte, le volume "BASESAV" apparaît comme suit :

Vol: basesav	Nbfic: 63/10	Lg.Gr: 4	Occup: 608/50	
EXEMPLE-S	EXEMPLE-T	OUVREF-A	OUVREF-S	PASS01-S
PASS02-S	REVBIM-S	REVMEN-S	REVTRI-S	TEXT-S

Vous remarquerez que les fichiers présents sur le volume destination n'ont pas été détruits. Les caractéristiques du volume destination ne sont pas modifiées. Le nom du volume, la taille des granules, et la longueur du catalogue restent invariants.

Le même commande avec l'option "RZ" :

—> CP,DV,f10,f11,RZ<rc>

donnerait le résultat suivant :

Vol: basesav	date: 25.11.84	Nbfic: 63/7	Lg.Gr: 4	Occup: 608/34
EXEMPLE-S	EXEMPLE-T	OUVREF-A	OUVREF-S	REVBIM-S
REVMEN-S	REVTRI-S			

Seuls les fichiers du volume origine subsistent sur le volume destination à l'issue de la copie.

Messages d'exécution

Lors de l'exécution de la commande de duplication de volume, le nom de chacun des fichiers en cours de copie est affiché successivement à l'écran. Si l'exécution se déroule sans incidents, le nom du dernier fichier est suivi du symbole "—" avant le retour à l'interpréteur de commandes ou à CP (mode répétitif). Les fichiers sont copiés dans l'ordre de leur présence dans le catalogue, c'est-à-dire dans l'ordre de leur création sur le volume origine.

* Destruction de volume :

Si l'option "RZ" est sélectionnée dans la commande d'appel et si l'unité destination contient effectivement un volume PROLOGUE validé, le message suivant apparaît :

Détruire le volume : nomvol (O-N) :—

où "nomvol" est le nom du volume présent sur l'unité destination.

Si vous répondez par OUI (O<rc> ou o<rc>), tous les fichiers sont DETRUIES avant la duplication. Toute autre réponse provoquera l'abandon de la commande de duplication et laissera intact le volume destination.

Avant de valider ce message par OUI, assurez-vous qu'aucun des fichiers présents sur le volume destination n'est significatif.

Messages d'erreur

En cours d'exécution de la commande de duplication de volume, une erreur fatale peut survenir, provoquant l'affichage d'un message d'erreur et l'abandon de la commande.

Si l'erreur survient lors de la copie d'un fichier, cette copie est invalide et le fichier destination est supprimé. La copie des fichiers précédents, dont les noms ont été affichés est cependant valide.

Après suppression des causes d'erreurs possibles, l'opération doit être relancée soit en ne recopiant que les fichiers restant à copier (inclus le fichier en cours de copie), soit en relançant complètement l'opération, ce qui est plus sûr.

Si l'erreur est une erreur système non particulière à CP, le libellé et le numéro de cette erreur sont affichés et vous vous reporterez à l'annexe B afin d'en connaître sa signification et les solutions envisageables.

Exemple :

—> CP,DV,fl0,fl1

Erreur fichier n° 54 : structure volume incorrecte

—> .

L'unité destination référencée par votre commande, ici "FL1" ne supporte pas un volume PROLOGUE d'un format correct.

Solution :

Assurez-vous qu'il n'y a pas de confusion sur la disquette introduite ou qu'elle n'a pas été détruite par une manipulation malencontreuse. Si c'est le cas, il est nécessaire de recréer un volume PROLOGUE sur cette disquette, à l'aide de la commande "—> CP,CV,fl1,....",

Si l'erreur est une erreur propre à la fonction DV de l'utilitaire CP, un message d'erreur particulier est affiché. Ci-dessous la signification de ces messages et les solutions possibles.

* "Fonction interdite"

Cause :

La fonction de duplication de volume ne vous est pas accessible avec le système PROLOGUE que vous utilisez.

Solution :

Pas de solution possible si cette interdiction est voulue pour des raisons de sécurité. Sinon reconfigurez votre système ou utilisez un système adéquat (reportez-vous au chapitre 7).

* “Erreur de commande n° 30 syntaxe incorrecte”

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction inexistant.
Ex : CP,SV,FL0,FL1<rc>
- nom d'unité invalide.
EX :,fl14,....
- absence d'un paramètre.
Ex : -> CP,DV,fl0,<rc>
- unités origine et destination identiques.
Ex : -> CP,DV,fl0,fl0<rc>
- mot-clé option incorrect.
Ex : -> CP,DV,fl0,fl1,RX<rc>

Et toutes autres erreurs de syntaxe.

Solution :

Corrigez la syntaxe de votre commande.

* “Ressource interdite”

Cause :

Une des unités référencées par votre commande n'est pas une ressource disque gérée par le Noyau de Gestion des Fichiers de PROLOGUE.

Ex : -> CP,DV,fl0,im0<rc>

“im0” désigne la ressource imprimante qui ne peut être l'objet d'une duplication de volume.

Solution :

Assurez-vous de la syntaxe de votre commande et de la bonne compréhension des mnémoniques des périphériques.

Exemple :

- FL = disquettes 5 pouces
- MD = disque fixe 5 pouces
- IM = ressource imprimante

* “opér. interdite sur vol. copie”

Cause :

L'unité destination contient un volume PROLOGUE de type “COPIE”, obtenu à partir de l'utilitaire SV, et votre commande présente l'option “RZ”. Dans ce cas, un volume de type “COPIE” est protégé et l'opération est refusée.

* “Débordement de volume, suppres. de : nomfic”

Cause :

En cours de copie, le volume destination est saturé.

Deux raisons peuvent engendrer cette erreur :

- 1) La capacité du volume destination est inférieure à celle du volume origine et ne peut contenir l'ensemble des fichiers.
- 2) Le volume destination contient préalablement des fichiers et la place disponible est insuffisante.

“nomfic” est le nom du fichier en cours de recopie lorsque l'erreur survient. Ce fichier est supprimé en destination car sa duplication est incorrecte.

Le message suivant :

“pour continuer changer de volume et taper C, (A = abandon) :”

vous invite soit à abandonner l'opération de duplication à ce stade (réponse “A” ou “a”), dans ce cas, tous les fichiers dont la copie a précédé celle de “nomfic” sont correctement dupliqués, le fichier “nomfic” lui-même ne l'étant pas, soit à poursuivre l'opération sur un autre volume PROLOGUE que vous aurez créé préalablement et que vous mettrez en place sur l'unité destination avant de poursuivre la duplication en validant par la réponse “C” ou “c”.

Solution :

Utilisez un volume destination de capacité suffisante ou utilisez "RZ" afin de supprimer préalablement les fichiers présents sur le volume destination.

- * "Débordement des extensions d'un fich., suppres. de : nomfic"

Cause :

Le nombre de blocs nécessaires à la recopie d'un fichier sur le volume destination est supérieur au nombre maximum de blocs d'un fichier de type 9F qui est de 18 ou avoir interdit plus de 18 blocs d'extensions (cf. ch. 7). "nomfic" est le nom du fichier en cours de duplication lorsque l'erreur survient. La copie de ce fichier est supprimée sur le volume destination car elle est incomplète.

Le message :

"pour continuer changer de volume et taper C, (A = abandon) :"

vous invite soit à abandonner l'opération de duplication à ce stade (réponse "A" ou "a"), dans ce cas, tous les fichiers dont la copie a précédé celle de "nomfic" sont correctement dupliqués, le fichier "nomfic" lui-même ne l'étant pas, soit à poursuivre l'opération sur un autre volume PROLOGUE que vous aurez préalablement créé et que vous mettrez en place sur l'unité destination avant de poursuivre la duplication en validant par la réponse "C" ou "c".

Solution :

Utilisez un volume destination de plus grande capacité ou définissez une taille de granules plus élevée sur le volume destination.

- * "Catalogue saturé, suppres. de : nomfic"

Cause :

En cours de copie, le catalogue destination est saturé. Le nombre effectif de fichiers créés sur le volume destination est égal au nombre maximum de fichiers que peut supporter ce volume. "nomfic" désigne le nom du fichier en cours de création sur le volume destination.

Le message suivant :

“pour continuer changer de volume et taper C,(A = abandon) :”

vous invite soit à abandonner l'opération de duplication à ce stade (réponse “A” ou “a”), dans ce cas tous les fichiers précédant “nomfic” sont correctement dupliqués, “nomfic” lui-même ne l'étant pas, soit à poursuivre l'opération sur un autre volume PROLOGUE que vous aurez préalablement créé et que vous mettrez en place sur l'unité destination avant de poursuivre la copie en validant par la réponse “C” ou “c”.

Solution :

La longueur du catalogue du volume destination est insuffisante. Pour l'étendre il est nécessaire de recréer un volume PROLOGUE avec la commande.

—> CP,CV,...

mais ATTENTION tous les fichiers présents sur ce volume seront supprimés.

Remarques et conseils

* Usage de la fonction CP,DV

Cette fonction permet la sauvegarde de l'ensemble des fichiers d'un volume. Pour un bon fonctionnement, le volume destination doit avoir des caractéristiques au moins égales à celles du volume source.

L'intérêt de cette fonction est de ne recopier que les informations utiles d'un volume, ce qui peut apporter un gain de temps important par rapport à une opération de duplication faite avec l'utilitaire CPS.

De plus les fichiers dupliqués sur le volume destination sont réorganisés. Le nombre de blocs les constituant est minimisé et les temps d'accès sont donc réduits.

* Taille des fichiers

La taille des fichiers n'est connue par le système qu'au granule près. Pour un fichier, un nombre entier de granules est donc recopié. Si la taille des granules destination est inférieure à celle des granules du volume origine, une perte de place peut survenir.

* Position des fichiers fixes

Certains fichiers, comme le fichier "SYSTEME" doivent être en position fixe sur le volume, afin d'être chargeables par l'ordinateur lors de l'initialisation. Lors d'une duplication de volume, la position des fichiers sur le volume destination peut varier. Il faut donc s'assurer de la bonne implantation des fichiers par la commande "`-> /,GR`".

Cependant, si le fichier "SYSTEME" est le premier créé sur un volume à partir du granule 16, lors de la duplication du volume, le fichier "SYSTEME" sera également le premier créé sur le volume destination, et si les volumes ont des caractéristiques identiques, sa position sera invariante.

* Option "RZ"

Lorsque la duplication se poursuit sur un ou plusieurs volumes supplémentaires, si l'option "RZ" est présente dans la commande d'appel, elle n'est pas propagée sur les volumes suivants. En conséquence, ceux-ci doivent être préalablement créés à l'aide de la commande "`CP,CV,....`".

4.2.2.4. Duplication d'un ou plusieurs fichiers (DF)

Présentation

Cette commande permet la duplication sélective d'un fichier vers un autre fichier ou vers la ressource console ou imprimante.

L'origine de la duplication peut être, soit un fichier unique, soit un groupe de fichiers appartenant à la même unité et sélectionnés par l'usage des caractères de substitution ("*" ou "?"), dans la commande d'appel.

La destination de la duplication peut être, soit un fichier unique situé sur le même volume ou un volume différent, soit une désignation d'unité qui peut être une unité de type disque ("FL" par exemple) ou l'unité implicite console ("CO") ou imprimante ("LO") de votre poste.

Si la destination est une désignation de fichier, l'origine de la duplication est nécessairement un fichier unique. Dans ce cas, la commande de duplication permet la recopie d'un fichier sur la même unité ou sur une unité différente. Si les unités origine et destination sont identiques, les désignations de fichiers (nom + suffixe) doivent être différentes. Si l'unité destination est l'unité console ("CO") ou imprimante ("LO"), le fichier ou le groupe de fichiers origine doivent être de type source, c'est-à-dire être éditables par un quelconque éditeur de textes. Dans le cas où des informations de type binaire sont envoyées sur ces unités, les résultats obtenus sont fantaisistes et aléatoires.

Commande d'appel

—> [unité1.]CP,DF,desfic1,[DEST=] $\left\{ \begin{array}{l} \text{unité2} \\ \text{desfic2} \\ \text{CO} \\ \text{LO} \end{array} \right\} <\text{rc}>$

où :

unité1 indique l'unité sur laquelle l'utilitaire CP est recherché. Si "unité1" est absent, CP est recherché sur l'unité implicite programme de votre poste.

CP nom du fichier binaire correspondant ayant comme suffixe "X".

- DF mot-clé spécifiant la fonction de duplication de fichiers.
- desfic1 désignation du fichier origine de la duplication (la désignation d'un groupe de fichiers d'une même unité se fait par l'usage du caractère de substitution dans la désignation du fichier). Si ce fichier comporte une clé d'accès lecture, elle doit être fournie dans la commande.
- DEST= mot-clé optionnel précisant la nature du paramètre suivant, destination de la duplication.

Une et une seule des options suivantes doit être choisie :

- unité2 unité destination de la duplication.
Le ou les fichiers sélectionnés par desfic1 sont dupliqués sur unité2 suivant une désignation identique : même nom, même suffixe, même clés d'accès. "unité2" doit être forcément différent de l'unité origine définie par "desfic1".

Si le fichier dupliqué existe préalablement sur "unité2", il est supprimé et remplacé par le fichier origine à la condition qu'il soit défini sans clés d'accès ou que ses clés d'accès soient identiques à celles du fichier origine.

Si les unités sont identiques, les désignations des fichiers doivent être différentes par le nom ou le suffixe.

Si le fichier destination existe préalablement, il est supprimé avant la duplication. S'il comporte une clé d'accès écriture, elle doit être indiquée dans la commande d'appel afin d'autoriser la suppression du fichier.

Si le fichier destination n'existe pas, les clés d'accès du fichier origine lui sont attribuées.

Si le suffixe du fichier est omis, le suffixe du fichier origine lui est attribué.

CO Mot-clé désignant l'unité console de votre poste comme unité destination de la duplication. Le contenu du fichier ou du groupe de fichiers désignés par "desfic1" est affiché sur l'écran de votre poste. Ce contenu doit être donc directement affichable, c'est-à-dire de type source.

L'option "CO" est l'option prise par défaut si le paramètre destination est omis de votre commande. A tout moment, l'affichage sur l'écran peut être suspendu à l'aide de la touche "SORTIE" (ou ESC) de votre clavier. Une deuxième frappe de cette touche permet de relancer l'affichage. La frappe successive de la touche "SORTIE" et de la touche "<rc>" (retour chariot) permet l'abandon de l'affichage du fichier courant et le passage à l'affichage du fichier suivant ou le retour à l'interpréteur de commandes ou à CP (mode répétitif) si le fichier courant est unique ou le dernier de la commande.

LO mot-clé désignant l'unité implicite d'impression de votre poste comme unité destination de la duplication. Le contenu du fichier ou du groupe des fichiers désignés par "desfic1" est édité sur l'imprimante implicite de votre poste.

Ce contenu doit être donc directement affichable, c'est-à-dire de type source.

Si le "SPOULEUR" est actif (voir paragraphe 4.2.9. "Gestion des états à imprimer"), le résultat sera pris en charge par le "SPOULEUR", préalablement à son impression.

A tout moment, l'impression peut être suspendue par la frappe de la touche "SORTIE" (ou ESC) de votre clavier. Une deuxième frappe permet la reprise de l'édition. La frappe successive des touches "SORTIE" et "<rc>" (retour chariot) provoque l'abandon de l'impression du fichier suivant ou le retour à l'interpréteur de commandes ou à CP (mode répétitif) si ce fichier est unique ou le dernier de la commande.

Exemple 1 :

Vous voulez dupliquer le fichier EXEMPLE-S sur le volume "BASE0001" situé sur l'unité implicite fichier de votre poste en lui donnant un autre nom : "COPEX-S", sur le même volume.

Utilisez la commande suivante :

```
-> CP,DF,EXEMPLE-S,COPEX<rc>
```

A la fin des opérations, le volume "BASE0001" contient les deux fichiers "EXEMPLE-S" et "COPEX-S".

Exemple 2 :

```
-> CP,DF,*-X,fl1<rc>
```

Tous les fichiers présents sur l'unité implicite fichier de votre poste dont le nom est quelconque et le suffixe égal à "X" sont dupliqués sur l'unité "FL1", différente de l'unité origine, suivant un nom, un suffixe et des clés d'accès identiques.

Exemple 3 :

```
-> CP,DF,fl0.*-S,LO<rc>
```

Tous les fichiers présents sur l'unité "FL0" dont le suffixe est "S" sont imprimés sur l'unité d'impression implicite de votre poste.

Exemple 4 :

```
-> CP,DF,*-*,fl1<rc>
```

Cette commande duplique l'ensemble des fichiers du support implicite utilisateur vers l'unité "FL1". Elle est identique à la commande de duplication de volume :

```
-> CP,DV,fl0,fl1<rc>
```

Exemple 5 :

```
-> CP,DF,prog-src:ab,DEST=fl1.savprog: xy<rc>
```

Le fichier "PROG-SRC" dont la clé d'accès en lecture est "ab" est dupliqué sur le fichier "SAVPROG-SRC" sur "FL1" dont la clé d'accès écriture est "xy". La clé d'accès lecture du fichier origine n'est pas dupliquée.

Messages d'exécution

La commande de recopie d'un groupe de fichier affiche au fur et à mesure les noms des fichiers concernés, l'affichage ayant lieu au début de la duplication du fichier. La commande se termine normalement par l'affichage d'un tiret. Par contre, si votre commande contient le nom d'un fichier spécifique, seul le "—" sera affiché. Après l'affichage de ce dernier le contrôle est rendu à l'interpréteur ou à CP si vous êtes en mode répétitif.

Exemple 1 :

```
--> CP,DF,FICH01-S,FL0 --  
--> .
```

Cette commande représente la copie d'un seul fichier. L'affichage du tiret derrière la commande indique que la copie s'est effectuée sans incident.

Exemple 2 :

```
--> CP,DF,FIC*-S,FL0  
FICH01-S      FICH02-S      --  
--> .
```

La commande, décrite ci-dessus, représente la copie d'un groupe de fichiers. Au fur et à mesure que les fichiers ont été recopiés, leur nom est affiché avec le suffixe. Les clés d'accès, si elles existent, ne sont pas affichées.

Messages d'erreur

En cours d'exécution de la commande de duplication de fichiers, une erreur fatale peut survenir provoquant l'affichage d'un message d'erreur et l'abandon de la commande.

Si l'erreur survient lors de la copie d'un fichier, cette copie est invalide et le fichier destination est supprimé. Dans le cas d'une duplication d'un groupe de fichiers, seul le fichier correspondant au dernier nom affiché peut être mal copié.

Après suppression des causes d'erreurs possibles l'opération doit être relancée soit en ne recopiant que les fichiers restant à copier (en incluant le fichier en cours de copie), soit en relançant complètement l'opération, ce qui est plus sûr.

Si l'erreur est une erreur système non particulière à CP, le libellé et le numéro de cette erreur sont affichés et vous vous reporterez à l'annexe B afin d'en connaître sa signification et les solutions envisageables.

Exemple :

—> CP,DF,*—S,fl1
. Erreur fichier n° 54 : structure volume incorrecte
—> .

L'unité destination référencée par votre commande, ici "FL1" ne supporte pas un volume PROLOGUE d'un format correct.

Solution :

Assurez-vous qu'il n'y a pas de confusion sur la disquette introduite ou qu'elle n'a pas été détruite par une manipulation malencontreuse. Si c'est le cas, il est nécessaire de recréer un volume PROLOGUE sur cette disquette, à l'aide de la commande

"—> CP,CV,fl1,....".

Si l'erreur est une erreur propre à la fonction DF de l'utilitaire CP, un message d'erreur particulier est affiché. Ci-dessous la signification de ces messages et les solutions possibles.

* “Erreur de commande n° 30 syntaxe incorrecte”

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction inexistant.
Ex : -> CP,SV,F1,FL1<rc>
- nom d'unité invalide.
Ex : ...,fl14,...
- mot-clé option incorrect.
Ex : -> CP,DF,*-S,fl1,RZ<rc>
- désignation de fichier incorrecte.
Ex : -> CP,DF,AZERTYUIOP,FL1<rc> ; le nom de fichier fait plus de 8 caractères.

Et toutes autres erreurs de syntaxe.

Solution :

Corrigez la syntaxe de votre commande.

* “Débordement de volume, suppres. de : nomfic”

Cause :

En cours de copie, le volume destination est saturé.

“nomfic” est le nom du fichier en cours de recopie lorsque l'erreur survient. Ce fichier est supprimé en destination car sa duplication est incorrecte.

Le message suivant :

“pour continuer changer de volume et taper C,(A = abandon) :”

vous invite soit à abandonner l'opération de duplication à ce stade (réponse “A” ou “a”), dans ce cas, tous les fichiers dont la copie a précédé celle de “nomfic” sont correctement dupliqués, le fichier “nomfic” lui-même ne l'étant pas, soit à poursuivre l'opération sur un autre volume PROLOGUE que vous aurez créé préalablement et que vous mettrez en place sur l'unité destination avant de poursuivre la duplication en validant par la réponse “C” ou “c”.

Solution :

Utilisez un volume destination de capacité suffisante. Mais ceci n'est valable que si la somme des espaces alloués pour vos fichiers à recopier est inférieure à l'espace libre maximum possible sur le type de support de destination.

* "Débordement des extensions d'un fich., suppres. de : nomfic"

Cause :

Le nombre de blocs nécessaires à la recopie d'un fichier sur le volume destination est supérieur au nombre maximum de blocs d'un fichier de type 9F qui est de 18.

"nomfic" est le nom du fichier en cours de duplication lorsque l'erreur survient. La copie de ce fichier est supprimée sur le volume destination car elle est incomplète.

Le message :

"pour continuer changer de volume et taper C,(A = abandon) :"

vous invite soit à abandonner l'opération de duplication à ce stade (réponse "A" ou "a"), dans ce cas, tous les fichiers dont la copie a précédé celle de "nomfic" sont correctement dupliqués, le fichier "nomfic" lui-même ne l'étant pas, soit à poursuivre l'opération sur un autre volume PROLOGUE que vous aurez créé préalablement et que vous mettrez en place sur l'unité destination avant de poursuivre la duplication en validant par la réponse "C" ou "c".

Solution :

Utilisez un volume destination de plus grande capacité ou définissez une taille de granules plus élevée sur le volume destination.

* “Catalogue saturé, suppres. de : nomfic”

Cause :

En cours de copie, le catalogue destination est saturé. Le nombre effectif de fichiers créés sur le volume destination est égal au nombre maximum de fichiers supportés par ce volume.

“nomfic” désigne le nom du fichier en cours de création sur le volume destination.

Le message suivant :

“pour continuer changer de volume et taper C,(A = abandon) :”

vous invite soit à abandonner l’opération de duplication à ce stade (réponse “A” ou “a”), dans ce cas, tous les fichiers dont la copie a précédé celle de “nomfic” sont correctement dupliqués, le fichier “nomfic” lui-même ne l’étant pas, soit à poursuivre l’opération sur un autre volume PROLOGUE que vous aurez préalablement créé et que vous mettrez en place sur l’unité destination avant de poursuivre la duplication en validant par la réponse “C” ou “c”.

Solution :

La longueur du catalogue du volume destination est insuffisante. Pour l’étendre il est nécessaire de recréer un volume PROLOGUE avec la commande —> CP,CV...

mais ATTENTION tous les fichiers présents sur ce volume seront supprimés.

* “Abs. nom de fichier”

Cause :

La commande d’appel ne contient pas de descripteur de fichier en origine.

Exemple :

—> CP,DF,MD0,FL0<rc>

Solution :

Refrappez la commande avec la désignation du fichier, ou du groupe de fichiers, choisi.

Remarques et conseils

* Usage de la fonction CP, DF

Cette fonction permet la sauvegarde d'un groupe de fichiers d'un volume ou la duplication d'un fichier sous un autre nom. Son intérêt est qu'elle ne dépend pas de la structure interne du fichier. Celui-ci est vu comme un ensemble de secteurs et donc tous les types de fichiers (source, programmes B.A.L., utilitaires,...) peuvent être dupliqués. De plus les fichiers dupliqués sur le volume destination sont réorganisés suivant l'espace disponible. Le nombre de blocs les constituant peut dans certains cas être minimum et les temps d'accès sont donc réduits.

* Taille des fichiers

La taille des fichiers n'est connue par le système qu'au granule près. Pour un fichier un nombre entier de granules est donc recopié. Si la taille des granules destination est inférieure à celle des granules du volume origine, une perte de place peut survenir.

* Position des fichiers fixes

Certains fichiers, comme le fichier "SYSTEME" doivent être en position fixe sur le volume, afin d'être chargeables par l'ordinateur lors de l'initialisation. Lors d'une duplication de fichiers, la position de ceux-ci sur le volume destination peut varier. Il faut donc s'assurer de la bonne implantation des fichiers par la commande " $\rightarrow$ /,GR", surtout si vous souhaitez charger votre système à partir du volume que vous avez utilisé comme volume de destination.

4.2.2.5. Création d'un fichier (CF)

Présentation

Cette fonction permet la création d'un fichier sur un volume PROLOGUE. Le nom du fichier sera enregistré dans le catalogue, mais aucun espace disque ne sera réservé. Ce n'est que lors de la première écriture de données au sein de ce fichier qu'un ou plusieurs granules lui seront alloués.

Commande d'appel

—> [unité1.]CP,CF,desfic1 <rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire CP est recherché. Si "unité1" est absent, CP est recherché sur l'unité implicite programme de votre poste.

CP nom du fichier binaire correspondant ayant comme suffixe "X".

CF mot-clé de la fonction de création de fichier.

desfic1 désignation du fichier à créer.
Voir les exemples ci-dessous.

Exemple :

—> CP,CF,FL0.FICH01-ASG:ABCD<rc>

Cette commande crée sur le volume se trouvant sur l'unité "FL0", un fichier dont le nom est "FICH01", de suffixe "ASG", dont les clés d'accès en lecture sont "AB" et les clés d'accès en écriture "CD". Si vous examinez le catalogue de votre volume au moyen de l'utilitaire "/", vous obtiendrez :

—> /,FL0,GR

Catalogue du volume Version 3.4d

Copyright (c) 1983 BULL Micral

Vol: TSTCP date: 24.10.1984 Nbfic: 63/1 Lg.Gr: 16 Occup: 152/2

FICH01-ASG : 0 :

—> .

On peut remarquer la taille du fichier qui est de 0 granule.

Messages d'exécution

Lors de l'exécution de la commande de création de fichier, aucun message n'est affiché. Mais lorsque la création est terminée, un "—" est affiché indiquant que la création de ce fichier s'est correctement passée. Puis, le contrôle est rendu à l'interpréteur ou à CP si vous êtes en mode répétitif.

Messages d'erreur

En cours d'exécution de la commande de création de fichiers, une erreur fatale peut survenir provoquant l'affichage d'un message d'erreur et l'abandon de la commande.

Si l'erreur survient lors de la création d'un fichier, le fichier n'est pas créé. Après suppression des causes d'erreurs possibles l'opération doit être relancée.

Si l'erreur est une erreur système non particulière à CP, le libellé et le numéro de cette erreur sont affichés et vous vous reporterez à l'annexe B afin d'en connaître sa signification et les solutions envisageables.

Exemple :

—> CP,CF,FL1.FICH01-S

Erreur fichier n° 54 : structure volume incorrecte

—> —

L'unité "FL1", sur laquelle l'utilisateur a souhaité créer son fichier, ne supporte pas un volume PROLOGUE de format correct.

Solution :

Assurez-vous qu'il n'y a pas de confusion sur la disquette introduite ou qu'elle n'a pas été détruite par une manipulation malencontreuse. Si c'est le cas, il est nécessaire de recréer un volume PROLOGUE sur cette disquette, à l'aide de la commande "—> CP,CV,fl1,...".

Si l'erreur est une erreur propre à la fonction CF de l'utilitaire CP, un message d'erreur particulier est affiché. Ci-dessous la signification de ces messages et les solutions possibles.

* “Erreur de commande n° 30 syntaxe incorrecte”

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction inexistant.
Ex : -> CP,CR,FICH01 <rc>
- nom d'unité invalide.
Ex : ...,f14,...
- mot-clé option incorrect.
Ex : -> CP,CF,FICH01,RZ <rc>
- description de fichier incorrecte.
Ex : -> CP,CF,AZERTYUIOP ; le nom de fichier fait plus de 8 caractères.
- utilisation du caractère de substitution “*”.
Ex : -> CP,CF,FICH*-S <rc>

Et toutes autres erreurs de syntaxe.

Solution :

Corrigez la syntaxe de votre commande.

* “Erreur Fichier n° 49 Catalogue saturé”

Cause :

Au moment de la création de votre fichier, le catalogue du volume devant contenir celui-ci est saturé. Le nombre effectif de fichiers créés sur le volume est égal au nombre maximum de fichiers supportés par ce volume.

Solution :

La longueur du catalogue du volume est insuffisante. Pour l'étendre il est nécessaire de recréer un volume PROLOGUE avec la commande.

-> CP,CV...

mais ATTENTION tous les fichiers présents sur ce volume seront supprimés.

* "Abs. nom de fichier"

Cause :

La commande d'appel ne contient pas de descripteur de fichier en origine.

Exemple :

—> CP,CF,FL0<rc>

Solution :

Refrappez la commande avec le descripteur de fichier correct.

* "Erreur Fichier n° 41 déjà créé"

Cause

Le fichier que vous souhaitez créer existe déjà sur le volume.

Solution :

Après vous être assuré que le nom de fichier est correct et que le volume sur lequel vous souhaitez le créer l'est aussi, supprimez le fichier de même nom pour libérer l'espace disque occupé, puis relancez votre commande.

4.2.2.6. **Suppression d'un ou plusieurs fichiers (SF)**

Présentation

Cette fonction permet de supprimer un fichier, ou un groupe de fichiers sur un seul volume à la fois. Cette fonction est particulièrement dangereuse parce qu'irréversible. En effet, une fois détruit, il est impossible de récupérer le fichier. Avant de lancer la commande, vérifiez bien que le ou les fichiers à supprimer sont ceux que vous souhaitez voir disparaître, et surtout vérifiez que le volume monté est bien celui sur lequel vous souhaitez faire vos suppressions.

Commande d'appel

—> [unité1.]CP,SF,desfic1<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire CP est recherché. Si "unité1" est absent, CP est recherché sur l'unité implicite programme de votre poste.

CP nom du fichier binaire correspondant ayant comme suffixe "X".

SF mot-clé spécifiant la fonction de suppression de fichier.

desfic1 désignation du fichier à supprimer (la désignation d'un groupe de fichiers d'une même unité se fait par l'usage des caractères de substitution dans la désignation du fichier). Voir les exemples ci-dessous. Si ce fichier comporte des clés d'accès, elles doivent être fournies dans la commande.

Exemple 1 :

—> CP,SF,FL0.FICH01-ASG:ABCD<rc>

Cette commande supprime, du volume se trouvant sur l'unité "FL0", le fichier dont le nom est "FICH01", dont les clés d'accès en lecture sont "AB" et les clés d'accès en écriture "CD".

Exemple 2 :

—> CP,SF,FICH*-ASG<rc>

Cette commande supprime, du volume se trouvant sur votre unité implicite fichier, tous les fichiers dont le nom commence par "FICH", de suffixe "ASG", et qui n'ont aucune clé d'accès en lecture et en écriture.

Messages d'exécution

Lors de l'exécution de la commande avec un seul fichier spécifié, aucun message n'est affiché. Par contre, lorsqu'on utilise une syntaxe indiquant la suppression d'un groupe de fichiers, l'utilitaire CP affiche d'abord la liste des fichiers à supprimer, puis demande à l'utilisateur confirmation de cette suppression. Après une réponse affirmative (la réponse négative provoque un retour à l'interpréteur, ou à CP si vous êtes en mode répétitif, et aucun fichier n'est supprimé), CP commence la suppression des fichiers, affichant le nom de ceux-ci au fur et à mesure de leur suppression.

Dans tous les cas, une fois la suppression terminée, un "—" est affiché indiquant que la suppression de fichier s'est correctement passée. Puis, le contrôle est rendu à l'interpréteur ou à CP si vous êtes en mode répétitif.

Exemple 1 :

—> CP,SF,FL0.FICH01-S —

—> —

L'utilisateur a supprimé un seul fichier (pas de caractère indiquant un groupe de fichier). L'affichage du "—" indique que cette suppression s'est effectuée correctement. Le retour à l'interpréteur de commandes indique que CP n'était pas utilisé en mode répétitif.

Exemple 2 :

```
—> CP,SF,FL0.FICH*-S      Liste des fichiers à supprimer
FICH01-S      FICH02-S      FICH03-S      FICH04-S
Lancement de la suppression (O-N) : O
FICH01-S      FICH02-S      FICH03-S      FICH04-S
> cp,—
```

L'utilisateur a supprimé un groupe de fichiers (utilisation du caractère “*”) dont le nom commence par “FICH” et dont le suffixe est “S”. Après que CP ait visualisé les quatre fichiers correspondant à ce critère, CP a demandé la confirmation de la suppression en affichant le message “lancement de la suppression (O-N) :”. L'utilisateur ayant répondu par l'affirmative, CP a réaffiché le nom des fichiers au fur et à mesure qu'ils étaient supprimés puis a affiché un “—” indiquant que la suppression s'était correctement passée. Enfin, le retour s'est fait sous CP (affichage du message “> cp,”), l'utilisateur ayant lancé sa commande en mode répétitif.

Message d'erreur

En cours d'exécution de la commande de suppression de fichiers, une erreur fatale peut survenir provoquant l'affichage d'un message d'erreur et l'abandon de la commande.

Si l'erreur survient lors de la suppression d'un fichier, celui-ci n'est pas détruit. Dans le cas d'une syntaxe définissant un groupe de fichiers, la suppression des fichiers précédant celui sur lequel s'est produit l'erreur, et dont les noms ont été affichés, reste cependant valide.

Après avoir remédié aux causes d'erreurs possibles, l'opération doit être relancée afin de poursuivre la suppression du ou des fichiers restants.

Si l'erreur est une erreur système non particulière à CP, le libellé et le numéro de cette erreur sont affichés et vous vous reporterez à l'annexe B afin d'en connaître sa signification et les solutions envisageables.

Exemple 1 :

—> CP,SF,FICH53-S

Erreur Fichier n° 40 Fichier inexistant

> cp, —

Le fichier “FICH53-S” n'existe pas sur le volume qui se trouve sur votre unité implicite fichier.

Solution :

Assurez-vous que le volume qui se trouve sur votre unité implicite fichier est le bon et que le fichier que vous souhaitez supprimer est bien celui que vous avez frappé dans votre commande.

Exemple 2 :

```
-> CP,SF,FL0.FICH*-S      Liste des fichiers à supprimer  
FICH01-S   FICH02-S   FICH03-S   FICH04-S   FICH05-S
```

Lancement de la suppression (O-N) : O

```
FICH01-S      FICH02-S
```

Erreur Fichier n° 47 Clés d'Accès Invalides

```
-> CP,SF,FL0.FICH03-S:abcd -
```

```
-> CP,SF,FL0.FICH*-S      Liste des fichiers à supprimer
```

```
FICH04-S      FICH05-S
```

Lancement de la suppression (O-N) : O

```
FICH04-S      FICH05-S
```

```
-> -
```

Dans la première commande, la suppression du troisième fichier a été refusée, les clés d'accès n'ayant pas été spécifiées. L'utilisateur a alors supprimé ce fichier en précisant les clés d'accès associées. Enfin, il a relancé la suppression des autres fichiers en relançant la commande initiale.

Si l'erreur est une erreur propre à la fonction SF de l'utilitaire CP, un message d'erreur particulier est affiché. Ci-dessous la signification de ces messages et des remèdes possibles.

* "Erreur de commande n° 30 syntaxe incorrecte"

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction inexistant.

Ex : -> CP,SF,F1,FL1<rc>

- nom d'unité invalide.

Ex : ...,fl4,...

- description d'un fichier incorrecte

Ex : -> CP,SF,AZERTYUIOP<rc> le nom de fichier fait plus de 8 caractères

Et toutes autres erreurs de syntaxe.

Solution :

Corriger la syntaxe de votre commande.

* “Abs. nom de fichier”

Cause :

La commande d'appel ne contient pas la désignation d'un fichier en origine.

Exemple :

—> CP,SF,FL0<rc>

Solution :

Refrappez la commande avec la désignation du fichier, ou du groupe de fichiers, choisi.

* “Aucun fichier !”

Cause :

Aucun fichier ne répond à la désignation définie dans la description du groupe de fichiers au sein de la commande.

Exemple :

—> CP,SF,FL0.FICH* Liste des fichiers à supprimer

Aucun fichier ! —

—>

Solution :

Assurez-vous que vous adressez le bon groupe de fichiers, sur le bon support et sur la bonne unité.

Remarques et conseils

* Usage de la fonction CP,SF

Cette fonction permettant donc de supprimer un ou plusieurs fichiers d'un même volume, restitue immédiatement les espaces occupés par ces fichiers. Cette commande est la seule qui permette de détruire un fichier, de plus elle est IRREVERSIBLE.

* Utilisation des clés d'accès dans la commande

Tous les fichiers ayant des clés d'accès en écriture ne pourront être supprimés que si ces clés sont spécifiées dans la commande mais alors les fichiers n'en ayant pas ne pourront être supprimés. Ainsi donc, la commande

"CP,SF,FICH*—S:ABCD<rc>"

ne pourra supprimer ni le fichier "FICH01-S", ni "FICH02-S:AB".

4.2.2.7. Modification du nom d'un fichier (RF)

Présentation

Cette fonction permet de modifier le nom d'un fichier, son suffixe ou ses clés d'accès. En aucun cas, elle n'affecte l'emplacement du fichier, ni son contenu.

Commande d'appel

—> [unité1.]CP,RF,desfic1,desfic2<rc>

où :

- | | |
|---------|--|
| unité1 | indique l'unité sur laquelle l'utilitaire CP est recherché. Si "unité1" est absent, CP est recherché sur l'unité implicite programme de votre poste. |
| CP | nom du fichier binaire correspondant ayant comme suffixe "X". |
| SF | mot-clé spécifiant la fonction de suppression de fichiers. |
| desfic1 | désignation du fichier origine à renommer. |
| desfic2 | nouvelle désignation du fichier. |

Exemple 1 :

—> CP,RF,FICH01-S,FICH01-LST<rc>

Avec cette commande, l'utilisateur souhaite modifier le suffixe du fichier "FICH01".

Exemple 2 :

—> CP,RF,FICH01-S,FICH01-S:ABCD<rc>

L'utilisateur ici souhaite ajouter les clés de protection "AB" en lecture et "CD" en écriture.

Exemple :

—> CP,RF,FICH01-S:ABCD,FICH01:AB<rc>

L'utilitaire souhaite supprimer les clés de protection en écriture "CD".

Messages d'exécution

Lors de l'exécution de la commande, aucun message n'est affiché. Une fois le fichier renommé, un "—" est affiché indiquant que la suppression de fichier s'est correctement passée. Puis, le contrôle est rendu à l'interpréteur ou à CP si vous êtes en mode répétitif.

Exemple :

```
—> CP,RF,FICH01-LST,FICH02 —  
—> _
```

Le fichier "FICH01-LST" a été renommé en "FICH02". L'opération s'est effectuée correctement puisque le "—" a été affiché. On remarquera que le nouveau nom de fichier n'a pas de suffixe, puisque l'utilisateur n'en a spécifié aucun ("CP,RF" n'impose aucun suffixe par défaut).

Message d'erreur

En cours d'exécution de la commande (renommer un fichier), une erreur fatale peut survenir provoquant l'affichage d'un message d'erreur et l'abandon de la commande : il n'y a aucune modification du nom du fichier.

Si l'erreur est une erreur système non particulière à CP, le libellé et le numéro de cette erreur sont affichés et vous vous reporterez à l'annexe B afin d'en connaître sa signification et les solutions envisageables.

Si l'erreur est une erreur propre à la fonction RF de l'utilitaire CP, un message d'erreur particulier est affiché. Ci-dessous, la signification de ces messages et les remèdes possibles.

- * "Erreur de commande n° 30 syntaxe incorrecte

Cause :

La syntaxe de la commande d'appel est incorrecte pour diverses raisons telles que :

- mot-clé de fonction inexistant.
Ex : —> CP,RN,F1,F2<rc>
- nom d'unité invalide.
Ex : ...,fl14,...
- mot-clé option incorrect.
Ex : —> CP,RF,F1,F2,CF<rc>
- description de fichier incorrecte.
Ex : —> CP,RF,F1,AZERTYUIOP<rc> ; le nom de fichier fait plus de 8 caractères.

Et toutes autres erreurs de syntaxe.

Solution :

Corrigez la syntaxe de votre commande.

- * "Abs. nom de fichier"

Cause :

La commande d'appel ne contient pas l'ancien nom de fichier ou le nouveau.

Exemple :

—> CP,RF,MD0,MD0.FICH03<rc>

Solution :

Refrappez la commande avec la désignation du fichier choisi.

* "Erreur fichier n° 40 Fichier Inexistant"

Cause :

Le fichier à renommer n'existe pas sur votre support.

Exemple :

—> CP,RF,FICH04,FICH05

Erreur de fichier n° 40 Fichier Inexistant

—> _

Solution :

Assurez-vous que vous avez spécifié le bon fichier, sur le bon support, sur la bonne unité.

* "Erreur fichier n° 41 Fichier Déjà Créé"

Cause :

Le nouveau nom que vous souhaitez donner à votre fichier est déjà présent.

Exemple :

—> CP,RF,FICH01,FICH02

Erreur Fichier n° 41 Fichier Déjà Créé

—> _

Solution :

Assurez-vous avant de lancer votre commande que le nouveau nom de fichier n'existe pas sur le support.

* "Localis. du fich. erronée"

Cause :

Le support associé au descripteur désignant le fichier à renommer est différent de celui associé au nouveau nom.

Exemple :

—> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0

—> CP,RF,FL0.FICH02, FICH05

Local. du fich. erronée

—> _

Solution :

Refrapper la commande avec les noms de support identiques dans les deux descripteurs de fichier.

* "Erreur Fichier n° 47 Clés d'Accès Invalides"

Cause :

Les clés d'accès spécifiées dans un des descripteurs sont incorrectes.

Exemple :

—> CP,CF,FICH01:ABCD _

—> CP,RF,FICH01:AB,FICH01

Erreur Fichier n° 47 Clés d'Accès Invalides

—> _

Solution :

Refrapper la commande en spécifiant toutes les clés d'accès de lecture et d'écriture.

Remarques et conseils

Cette fonction permet de modifier le nom, le suffixe, les clés d'accès d'un fichier. Pour ces dernières, il convient de noter quelque part leur valeur, car aucune fonction du système ne permet de les lire. L'intérêt de renommer un fichier est aussi de libérer un nom de fichier sans détruire celui-ci. Un programme créant systématiquement un fichier, peut être gêné si celui-ci existe déjà. En renommant un fichier, on le sauvegarde et on évite qu'il soit altéré par le programme qui l'a créé.

4.2.3. VISUALISATION ET MISE A JOUR DE LA DATE : "DATE"

Présentation

L'heure et la date sont tenues à jour en permanence par le système. Lorsque vous chargez votre système, PROLOGUE prend par défaut une heure et une date que vous pourrez visualiser et éventuellement modifier au moyen d'un seul utilitaire : DATE.

L'utilitaire DATE peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (DATE est installable en mémoire mais non partageable). Cette installation se fait par la déclaration suivante dans le fichier source "SYSCONF-S" :

– DATE < INTR

Reportez-vous au chapitre 7 (Configuration de PROLOGUE) pour plus de détails.

Commandes d'appel

Première syntaxe :

–> [unité.]DATE<rc>

où :

unité indique l'unité sur laquelle l'utilitaire DATE est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire DATE est recherché sur l'unité implicite programme de votre poste.

DATE nom du fichier binaire correspondant ayant comme suffixe "X".

Deuxième syntaxe :

–> [unité.]DATE,jj/mm/aa hh:mn<rc>

où :

unité indique l'unité sur laquelle l'utilitaire DATE est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire DATE est recherché sur l'unité implicite programme de votre poste.

DATE nom du fichier binaire correspondant ayant comme suffixe "X".

jj/mm/aa définition de la nouvelle date (où jj représente le jour, mm le mois, aa l'année).

hh:mn définition de la nouvelle heure (où hh représente l'heure, mn les minutes).

Vous avez donc deux syntaxes possibles. L'une ou l'autre pourra être utilisée, indifféremment, pour modifier date et heure. Seule la première syntaxe sera utilisée pour la visualisation de la date et de l'heure.

La première syntaxe visualisera d'abord la bannière et le numéro de version de l'utilitaire puis la date et l'heure. Ensuite, l'utilisateur sera sollicité pour modifier ceux-ci. La frappe d'un <rc> laissera date et heure inchangées, et provoquera le retour à l'interpréteur de commandes. Pour modifier date et heure ou seulement la date ou seulement l'heure, l'utilisateur pourra, au moyen du déplacement du curseur, se placer sur le paramètre à modifier et le mettre à jour. Une fois les mises à jour de ces paramètres effectuées, la frappe du <rc> validera celles-ci et provoquera le retour à l'interpréteur de commandes.

Avec la seconde syntaxe, l'utilisateur entre directement derrière la virgule la nouvelle date et la nouvelle date et la nouvelle heure en respectant scrupuleusement leur mise en forme.

Exemple 1 :

—> DATE<rc>

Initialisation de la date et de l'heure Version 1.2a

Copyright (c) 1983 BULL Micral

(jj/mm/aa hh:mn) : 03/07/85 16:35

—> DATE<rc>

Initialisation de la date et de l'heure Version 1.2a

Copyright (c) 1983 BULL Micral

(jj/mm/aa hh:mm) : 03/09/85 17:52

—> _

Après la visualisation de la date (première commande), l'utilisateur a modifié le mois puis l'heure. Pour cela il a placé son curseur sur le 7, puis a frappé 9, puis a déplacé de nouveau son curseur sur le 6 pour frapper 7:52. On notera que les séparateurs (blanc, "/", ":") sont obligatoires et doivent être, soit frappés au clavier, soit sautés au moyen du curseur. Tout autre caractère pris comme séparateur provoque l'affichage d'un message d'erreur, après validation de la commande.

Remarque :

Un paramètre peut être modifié plusieurs fois avant de revenir à l'interpréteur. C'est la dernière modification qui sera prise en compte.

Message d'exécution

DATE n'envoie pas de message indiquant que la modification s'est correctement effectuée.

Message d'erreur

Pour toute erreur dans la tentative de modification d'un paramètre, l'utilisateur retournera une erreur de commande numéro 30.

Exemple 1 :

—> DATE<rc>

Initialisation de la date et de l'heure Version 1.2a

Copyright (c) 1983 BULL Micral

(jj/mm/aa hh:mm) : 03/07/85 16:35

Erreur de Commande n° 30 Syntaxe Incorrecte

—> _

Le caractère ":" n'est pas le bon séparateur de la zone date.

Exemple 2 :

—> DATE<rc>

Initialisation de la date et de l'heure Version 1.2a

Copyright (c) 1983 BULL Micral

(jj/mm/aa hh:mm) : 31/06/85 16:35

Erreur de Commande n° 30 Syntaxe Incorrecte

—> _

Il n'y a que 30 jours dans le mois de juin.

Exemple 3 :

—> DATE<rc>

Initialisation de la date et de l'heure Version 1.2a

Copyright (c) 1983 BULL Micral

(jj/mm/aa hh:mm) : 29/02/83 16:35

Erreur de Commande n° 30 Syntaxe Incorrecte

—>

Le mois de février n'a que 28 jours, car 1983 n'est pas une année bissextile.

4.2.4. VISUALISATION ET MISE A JOUR DES IMPLICITES "VPMI"

Présentation

L'utilitaire VPMI permet de modifier la liste des unités implicites attachées à votre poste. Ainsi vous pouvez changer votre unité implicite programme ; lorsque vous lancerez une commande d'appel pour activer un utilitaire, si celui-ci n'est pas résident et si aucune unité n'est spécifiée devant son nom dans la commande, alors il sera chargé par le système à partir du support nouvellement spécifié. De même vous pouvez modifier votre implicite fichier ; les fichiers spécifiés dans une commande d'utilitaire se verront chargés à partir du support se trouvant sur l'unité nouvellement spécifiée. Enfin, vous pouvez modifier votre implicite imprimante ; les états envoyés sur l'imprimante seront dirigés vers la nouvelle unité sélectionnée.

L'intérêt de VPMI est donc de modifier les unités implicites d'un poste en vue d'alléger les commandes et les spécifications des fichiers de la description de l'unité. Bien sûr, on peut toujours atteindre ceux qui ne se trouvent pas sur les unités implicites en spécifiant cette unité. Mais dans une exploitation, on recherchera toujours à éviter ces syntaxes lourdes et source d'erreur.

L'utilitaire VPMI peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (VPMI est installable en mémoire mais non partageable). Cette installation se fait par la déclaration suivante dans le fichier source "SYSCONF-S".

— VPMI < INTR

Reportez-vous au chapitre 7 (configuration des utilitaires résidents) pour plus de détails.

ATTENTION !

Dans certaines versions de l'utilitaire "VPMI", le libellé des unités implicites programme et fichier sont respectivement "unité implicite système" et "unité implicite utilisateur". Cette terminologie peu claire a été abandonnée car elle pouvait prêter à confusion entre des notions de poste et des notions de système.

Commandes d'appel

Première syntaxe :

—> [unité1.]VPMI[,LIS=LO]<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire VPMI est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire VPMI est recherché sur l'unité implicite programme de votre poste.

VPMI nom du fichier binaire correspondant ayant comme suffixe "X".

LIS=LO recopie sur imprimante des modifications faites sur les implicites.

Deuxième syntaxe :

—> [unité1.]VPMI[,IPRG=unité2][,IFIC=unité3][,IPRT=unité4]<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire VPMI est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire VPMI est recherché sur l'unité implicite programme de votre poste.

VPMI nom du fichier binaire correspondant ayant comme suffixe "X".

IPRG= mot-clé désignant l'implicite programme.

unité2 nouvelle unité implicite programme pour le poste.

IFIC= mot-clé désignant l'implicite fichier.

unité3 nouvelle unité implicite fichier pour le poste.

IPRT= mot-clé désignant l'implicite imprimante.

unité4 nouvelle unité implicite imprimante pour le poste.

Pour modifier vos implicites vous avez donc deux syntaxes possibles.

La première syntaxe visualisera d'abord des informations générales liées au système, telles que : la version du système, les différents types de périphériques existants ainsi que le nombre d'unités disponibles. Puis VPMI affichera les implicites liés à votre poste. A ce moment vous pourrez modifier vos implicites en répondant aux différents messages de VPMI. Si vous ne souhaitez pas modifier un implicite, frapper <rc> suffit. Le retour à l'interpréteur se fera en appuyant sur la touche ESC (ou SORTIE). Un caractère “—” s'affichera et vos implicites seront alors modifiés.

Le deuxième syntaxe permet de modifier directement les implicites sans visualisation de ceux-ci au préalable. L'affichage d'un “—” signifiera que la modification s'est bien passée.

Exemple 1 :

```
-> Prologue V2.3 Poste 0 Implicites : MD0,MD0,IM0
-> VPMI,IPRG=>FL0,IFIC=>FL1,IPRT=>IM1 <rc>
-> Prologue V2.3 Poste 0 Implicites : FL0,FL1,IM1
-> CP,DF,SYSCONF-S,LIS=>LO<rc>
-> —
```

Après l'exécution de VPMI, le poste 0 a comme implicite programme FL0, comme implicite fichier FL1 et comme implicite imprimante IM1. Lors de l'exécution de l'utilitaire qui suit, CP-X est chargé à partir de FL0 et le fichier SYSCONF-S sera lu sur FL1 pour être imprimé sur l'imprimante IM1.

Exemple 2 :

```
-> Prologue V2.3 Poste 0 Implicites : MD0,MD0,---
-> VPMI,IFIC=>FL0<rc>
-> Prologue V2.3 Poste 0 Implicites : MD0,FL0,---
-> —
```

Après l'exécution de VPMI, l'implicite fichier devient FL0. Les autres restent inchangés.

Exemple 3 :

```
-> Prologue V2.3 Poste 0 Implicites : MD0,MD0,IM0
-> VPML<rc>
    Visualisation et modification des implicites de poste
    Version 1.3b
    Copyright (c) 1983 BULL Micral
PROLOGUE V2.3
périphériques : MD FL IM
nombre d'unités : 1 2 1
implicite programme : MD0
implicite fichier : MD0
implicite imprimante : IM0

    Modifications :
implicite programme : MD0
implicite fichier : FL0
implicite imprimante : IM0
implicite programme : MD0
implicite fichier : FL1
implicite imprimante : -
-> Prologue V2.3 Poste 0 Implicites : MD0,FL1,IM0
-> -
```

Seul l'implicite fichier a été modifié. On remarquera, dans cet exemple, que l'utilisateur a modifié deux fois son implicite. C'est la dernière modification qui a été prise en compte.

Par ailleurs, on notera que si on frappe <rc> à une demande de modification VPML réaffiche l'implicite en cours.

Messages d'exécution

Seul l'affichage d'un "—" avant le retour à l'interpréteur signale à l'utilisateur que la modification s'est effectuée correctement.

Messages d'erreurs

VPMI vérifie la syntaxe et la cohérence des informations entrées par l'utilisateur. Un mot-clé inexistant ou erroné, la désignation d'une unité non présente sur le système provoquent l'affichage d'un message d'erreur. Mais attention, VPMI ne vérifie pas la logique de votre modification. Par exemple, si dans la modification de l'unité implicite programme de votre poste vous indiquez l'unité "IM0" (unité 0 imprimante) et que celle-ci existe réellement, VPMI ne renverra aucun message d'erreur, car il considère que la syntaxe est bonne et que cette unité est bien définie dans votre système.

* "Erreur Fichier n° 36 Fonction Non Supportée"

Ce message apparaît lorsqu'avec la syntaxe (1), on utilise l'option "LIS=LO" et que la fonction imprimante n'existe pas dans le système. Si vous souhaitez donc utiliser cette option, modifiez votre configuration (voir chapitre 7), puis rechargez votre système.

Exemple :

```
-> VPMI,LIS=LO
    Visualisation et modification des implicites de poste
    Version 1.3a
    Copyright (c) 1983 BULL Micral

    Erreur Fichier n° 56 Fonction Non Supportée
-> _
```

* "Erreur de Commande n° 30 Syntaxe Incorrecte"

Avec la syntaxe (2), toute erreur dans la commande, que ce soit dans la syntaxe ou la désignation des périphériques, provoquera l'affichage de l'erreur 30. Assurez-vous donc que le mot-clé utilisé est correct ou que la nouvelle unité spécifiée existe sur votre système.

Exemple 1 :

—> VPML,IPR=FL0

Erreur de Commande n° 30 Syntaxe Incorrecte

—> —

“IPR” n’est pas un mot-clé.

Exemple 2 :

—> VPML,IPRG=FL4

Erreur de Commande n° 30 Syntaxe Incorrecte

—> —

Le système n’a pas cinq unités de disques souples et “FL4” ne peut donc être atteint.

4.2.5. COPIE PHYSIQUE DE SUPPORT : "CPS"

Présentation

L'utilitaire CPS permet de recopier un support, ou une partie d'un support sur un autre support. Cette copie est une copie physique, c'est-à-dire qu'elle ne tient pas compte de l'organisation du volume ni de la structure logique du volume, des fichiers et des données. Ainsi donc, avec cet utilitaire seront recopiées les zones utilisées et celles qui sont libres.

L'utilitaire CPS ne peut pas être installé de manière permanente en mémoire. Il vous faut donc l'avoir sur un des supports de votre système. De plus, vous ne pourrez l'utiliser que si les paramètres de protection d'exploitation, spécifiés lors de la génération de votre système, vous y autorisent. Pour plus de précision sur ces protections d'exploitation et leur mise en œuvre, reportez-vous au chapitre 7 (Configuration de PROLOGUE).

4.2.5.1. Copie intégrale de support

Présentation

Le volume est reproduit dans sa totalité, sans modification ni de la structure ni de l'emplacement des fichiers. Ce type de copie vous permet, en particulier, de reproduire des supports dont le format est différent du format PROLOGUE.

Commande d'appel

—> [unité1.]CPS,unité2,unité3[.PM]<rc>

où :

- | | |
|--------|---|
| unité1 | indique l'unité sur laquelle l'utilitaire CPS est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire CPS est recherché sur l'unité implicite programme de votre poste. |
| CPS | nom du fichier binaire correspondant ayant comme suffixe "X". |
| unité2 | unité d'origine. Unité que l'on désire recopier. |
| unité3 | unité de destination. Unité sur laquelle sera faite la copie. Doit être de même type que l'unité d'origine. |
| PM | prémarquage de support se trouvant sur l'unité de destination. |

Exemple 1 :

—> FL0.CPS,FL0,FL1 détruire le volume : voltest (O/N) : O

L'utilitaire sera chargé à partir de l'unité FL0. Avant la recopie de l'unité FL0 vers l'unité FL1, CPS demandera à l'utilisateur l'autorisation de la destruction du volume de destination en affichant le nom de ce dernier. On remarquera, après l'exécution de cette commande, que le fichier "CP-X" est au même emplacement sur les deux unités et qu'il contient exactement le même nombre de blocs d'extension aux mêmes emplacements.

Exemple 2 :

—> CPS,FL0,FL1,PM prémarquage (0/n) ? : n

L'utilitaire sera chargé à partir de l'unité implicite programme. Avant d'exécuter la copie, il demande la confirmation du prémarquage du support de destination.

Une réponse négative provoque l'abandon immédiat de l'utilitaire, le volume de destination restant intact.

Messages d'exécution

En cours d'exécution, le nombre de pistes restant à copier est affiché. Ce nombre décroît en même temps que la copie progresse. Lorsque toutes les pistes ont été copiées, le nombre 0 est affiché et l'affichage d'un tiret vous signale que la copie est terminée et qu'elle s'est correctement effectuée.

Exemple :

—> CPS,FL0,FL1,PM prémarquage (o/n) ? : O

L'utilitaire sera chargé à partir de l'unité implicite programme. Avant d'exécuter la copie, il demande la confirmation du prémarquage du support de destination. Avec une réponse affirmative, nous verrons la ligne de commande évoluer comme suit :

—> CPS,FL0,FL1,PM prémarquage (0/n) ? : O prémarquage en cours

A ce moment le volume de destination est en prémarquage mais la copie n'a pas encore commencée. Puis :

—> CPS,FL0,FL1,PM prémarquage (o/n) ? : O prémarquage en cours 51

Là, nous sommes en cours de copie et 51 indique le nombre de pistes restant à copier. Puis :

—> CPS,FL0,FL1,PM prémarquage (o/n) ? : O prémarquage en cours —

—> —

Enfin, le nombre de pistes restant à transférer et l'affichage du tiret nous signalent que la copie s'est correctement effectuée.

Messages d'erreur

En plus des messages d'erreurs système sur les périphériques, l'utilitaire CPS émet quelques messages propres à son fonctionnement.

Exemple 1 :

—> CPS,MD0,FL0

Supports différents

L'utilitaire signale ici que les supports d'origine et de destination doivent être de même type.

Exemple 2 :

—> CPS,FL0,FL1

Erreur périphérique n° 3

Le système signale à l'utilisateur que le support de destination a un format qui ne peut être lu par le contrôleur. Il convient alors de vérifier les disquettes et, si cette erreur se produit sur le support de destination, de prémarquer celui-ci.

4.2.5.2. Copie partielle de support

Présentation

Vous pouvez copier quelques pistes, ou quelques secteurs, d'un support sur un autre support. Celui-ci peut être d'un type différent du support d'origine. Il conviendra de faire attention aux longueurs des pistes de chaque support. Un disque dur peut avoir des pistes de 32 secteurs alors que la disquette de destination aura des pistes de 16 secteurs. Dans ce cas, il faudra deux pistes d'une disquette pour recevoir une piste d'un disque.

Commande d'appel

-> CPS<rc>,<rc>

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

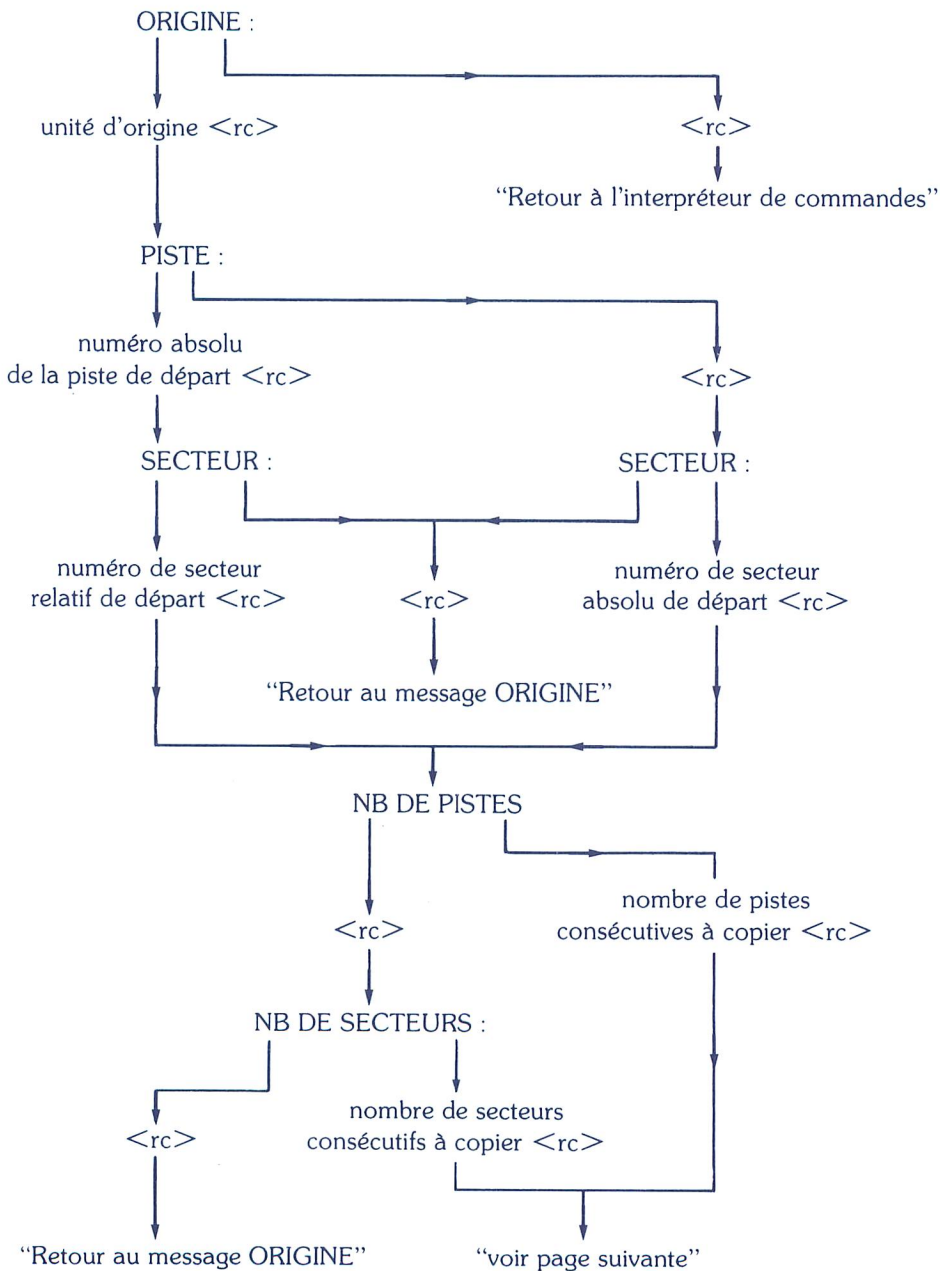
Origine :

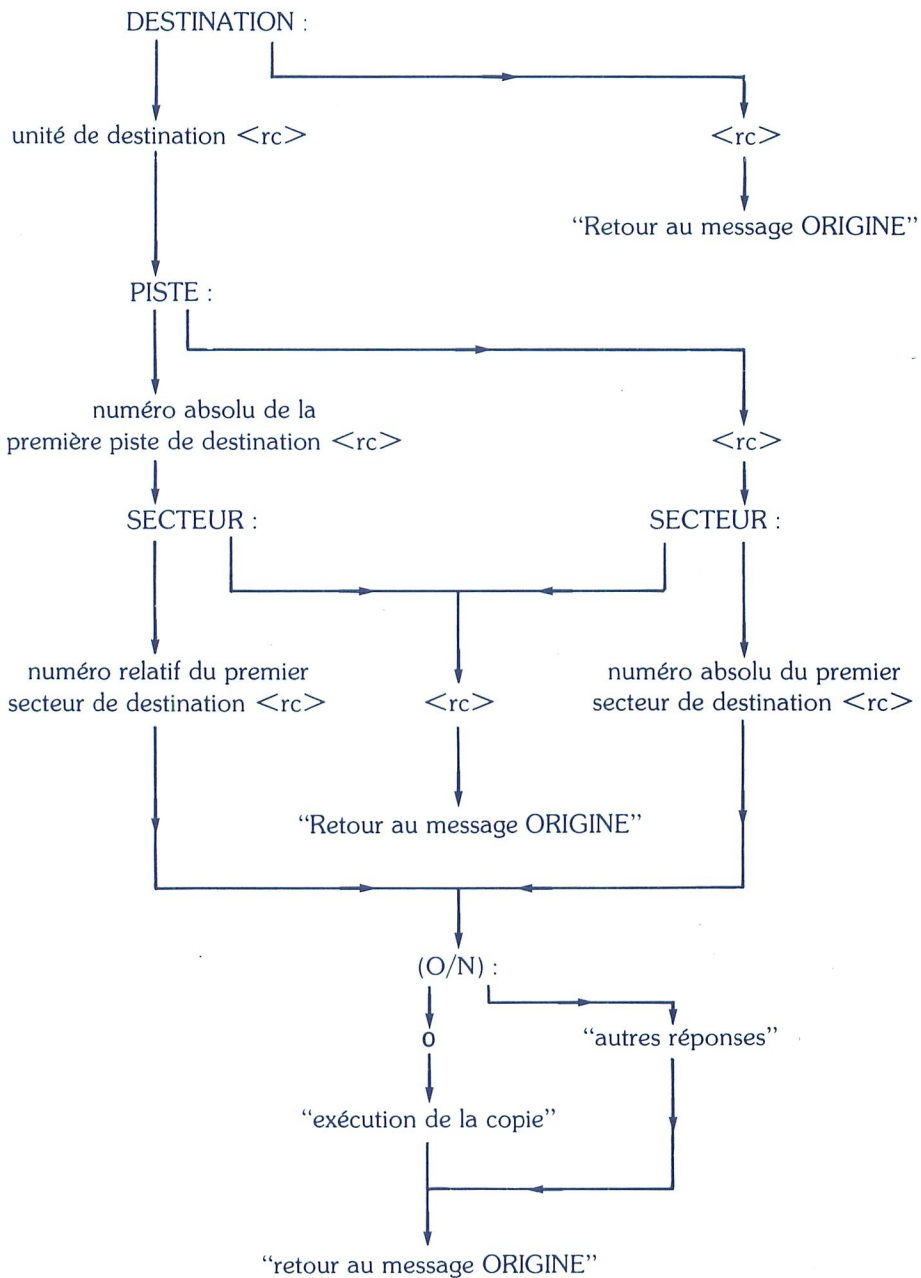
L'appel à la copie partielle se fait en frappant CPS suivi de deux retours-chariots. A partir de ce moment, s'instaure un dialogue entre la machine et l'utilisateur sur :

- l'unité d'origine,
- la localisation du début de la zone à transférer,
- la taille de ce transfert,
- l'unité de destination,
- la localisation du début de la zone de réception.

Certains des messages, lorsqu'ils ne sont plus nécessaires, sont effacés de l'écran et remplacés par le suivant. Une fois tous les paramètres entrés, le système demandera une confirmation.

Diagramme du dialogue entre l'utilisateur et la machine (en majuscules les messages du système, en minuscules les réponses de l'utilisateur, entre guillemets les commentaires).





Remarques

- Les numéros et les adresses peuvent être spécifiés en décimal ou en hexadécimal. Dans ce cas, le nombre doit être suivi du caractère "H".
- Les pistes et secteurs, lorsqu'ils sont désignés, doivent être numérotés à partir de zéro.

Exemple 1 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine:MD0 <rc> Piste : 0 <rc> Secteur : 0 <rc> Nb. de pistes : 76 <rc>
Destination : FL0 <rc> Piste : 0 <rc> Secteur : 0 <rc> (O/N) : 0 <rc> —

Origine : —

Nous avons eu ici la recopie des 76 premières pistes du disque dur MD0 sur les premières pistes du disque souple FL0. Cette recopie s'est passée sans incident, puisque le tiret a été affiché ainsi que le message "Origine" :".

Maintenant, nous pouvons refaire une copie en répondant aux différents messages qui apparaîtront, ou bien revenir à l'interpréteur de commandes en frappant la touche <rc>.

Avant de lancer une copie partielle, il convient de bien analyser le contenu à transférer et le contenu transféré, en regardant les caractéristiques des supports. Tout d'abord, ne pas confondre la taille d'un granule avec le nombre de secteurs par piste. Dans cet exemple, si les pistes du disque sont de 32 secteurs par piste et celles du disque souple de 16 secteurs par piste, il faut deux pistes du disque souple pour contenir une piste du disque dur. Ainsi pour notre copie, il aura fallu 152 pistes sur le disque souple. Ensuite, il convient de s'assurer que le dernier secteur peut être lu puis transféré. Ici, si la capacité du disque dur est de 10 Méga octets et celle du disque souple, de 608 koctets, il n'y a aucun problème. On peut même remarquer que le disque souple sera entièrement rempli.

Remarque

Avec la copie partielle, il est possible de faire, comme nous avons commencé à le faire avec cet exemple, une sauvegarde disque. Toutefois, il convient de s'armer de courage et de disques souples, car ce type de sauvegarde est une sauvegarde intégrale et donc les parties non utilisées sont aussi sauvegardées. Ce type de sauvegarde est donc fortement déconseillée pour les volumes faiblement remplis. De plus, la restauration ne pouvant se faire que par l'utilitaire CPS, il convient de présenter les disques souples dans le bon ordre, aucune vérification n'étant faite sur les données restaurées.

Exemple 2 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine : MD0 <rc> Piste : 0 <rc> Secteur : 0 <rc> Nb. de secteurs : 2432 <rc>

Destination : FL0 <rc> Piste : 0 <rc> Secteur : 0 <rc> (O/N) : O <rc> —

Origine : —

Cet exemple montre le transfert des 2432 premiers secteurs du disque dur MD0 vers les premiers secteurs du disque souple FL0. Nous pouvons remarquer que si les caractéristiques des supports sont les mêmes que dans l'exemple précédent, les deux exemples ont des résultats identiques. En effet, 2432 secteurs sur MD0, ce sont 76 pistes de 32 secteurs par piste.

Exemple 3 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine : FL0 <rc> Secteur : 0 <rc> Nb. de secteurs : 2431 <rc>

Destination : FL0 <rc> Secteur : 0 <rc> (O/N) : O <rc> —

Origine : —

Cet exemple est correct. Il est possible de recopier une partie d'un support sur ce même support, même si une partie de la zone d'origine est écrasée. Toutefois, le transfert, comme ici, ne se fait pas secteur par secteur, et donc le résultat ne sera pas l'initialisation de tout ou une partie du support. CPS travaille avec des tampons qui contiennent plusieurs secteurs. Donc, plusieurs secteurs sont lus puis écrits.

Messages d'exécution

Le tiret sera affiché après l'exécution de la copie. Dans tous les cas, le message "Origine : " sera ensuite affiché, pour une nouvelle copie. Si on frappe <rc>, on revient dans le contexte qui a lancé CPS, en général l'interpréteur de commandes.

Messages d'erreur

Exemple 1 :

```
-> CPS,  
    Copie physique de supports sectorisés Version 2.3a  
    Copyright (c) 1983 BULL Micral
```

```
Origine  : MD0  
Nom de support incorrect
```

```
Origine  : _
```

Ce type de support n'existe pas sur votre système. CPS le signale, puis affiche de nouveau le message "Origine : ".

Exemple 2 :

```
-> CPS,  
    Copie physique de supports sectorisés Version 2.3a  
    Copyright (c) 1983 BULL Micral
```

```
Origine  : MD9  
    Erreur de Commande n° 30 Syntaxe Incorrecte
```

```
Origine  : _
```

Ce type de support existe bien, mais pas l'unité 9. CPS le signale en affichant l'erreur système 30.

Exemple 3 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine : MD0 Piste : b12h Secteur : 0 Nb. de secteurs : 2432

Destination : FL0 Secteur : ABC

Paramètre incorrect

Origine : —

Tout nombre hexadécimal doit être suivi immédiatement de la lettre "H".
Ces nombres peuvent être écrits en majuscules ou minuscules.

Exemple 4 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine : MD0 Piste : FFFFFFFh

Nombre en entrée trop grand

Origine : —

Ce nombre trop grand est refusé par CPS, ne pouvant être traité. De plus,
votre support ne contient probablement pas autant de pistes.

Exemple 5 :

—> CPS,

Copie physique de supports sectorisés Version 2.3a

Copyright (c) 1983 BULL Micral

Origine : MD0 Piste : FFFh Secteur : 0 Nb. de secteurs : FFFFh

Destination : FL0 Secteur : 0 (O/N) : O

Erreur périphérique n° 4 Accès illégal

Origine : —

Les nombres entrés ne sont pas trop grands et donc n'ont pas été refusés par CPS. Par contre, un secteur n'a pu être lu parce qu'il n'existe pas physiquement : l'adresse demandée étant hors des limites du disque MD0. Avant de lancer une copie partielle, il convient donc de s'assurer que le premier secteur et le dernier secteur des unités d'origine et de destination peuvent être atteints.

Exécution différée de CPS :

—> CPS <rc>

PROLOGUE affichera

—> CPS,

Cette fonction est utile lorsque CPS n'est pas sur le volume à copier. On peut après l'affichage de la virgule, changer de support.

4.2.6. SAUVEGARDE DES FICHIERS : "SV"

Présentation

La commande SV permet d'effectuer la sauvegarde intégrale ou partielle d'un volume. Son intérêt est, contrairement aux utilitaires CP et CPS qui sont des utilitaires de duplication, la numérotation des volumes (avec une indication qui précise le dernier volume de sauvegarde) et la recopie des fichiers à cheval sur deux ou plusieurs volumes permettant la sauvegarde de très gros fichiers. En effet, avec l'utilitaire CP, il est impossible de recopier des fichiers dont la taille est supérieure à celle du support de sauvegarde. De plus, afin de ne pas utiliser les volumes de sauvegarde comme les autres volumes, un indicatif spécifique *COPIE* apparaît lors de la visualisation du contenu au moyen de l'utilitaire "/".

Une fois la sauvegarde effectuée, l'utilisation de CP pour récupérer ou dupliquer un fichier à partir d'un volume *COPIE* est fortement déconseillée car CP ne traite pas les fichiers à cheval sur deux volumes. Pour récupérer un fichier, il faudra donc OBLIGATOIREMENT utiliser l'utilitaire de restauration RT.

L'utilitaire SV peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (SV est installable en mémoire mais non partageable). Cette installation se fait par la déclaration suivante dans le fichier source. "SYSCONF-S" :

— SV < INTR

Reportez-vous au chapitre 7 (Configuration des utilitaires résidents) pour plus de détails.

Commande d'appel

—> [unité.]SV,nomfich,[DEST=]unité2[,PM]<rc>

où :

- unité1 indique l'unité sur laquelle l'utilitaire SV est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire SV est recherché sur l'unité implicite programme de votre poste.
- SV nom du fichier binaire correspondant ayant comme suffixe "X".
- nomfich désigne le fichier, le groupe de fichiers ou l'unité que vous souhaitez sauvegarder.
- DEST= ce mot désigne la destination et sert donc à différencier l'unité d'origine et l'unité de destination.
- unité2 indique l'unité de réception des fichiers à sauvegarder. Sur cette unité seront montés successivement les différents volumes nécessaires à la sauvegarde.
- PM cette option permet un prémarquage de chaque volume de destination avant la copie. A chaque changement de volume, SV demandera la confirmation du prémarquage. Si cette option n'est pas spécifiée dans votre commande, à chaque changement de volume sur l'unité de destination, vous devrez présenter des supports déjà prémarqués.

Exemple 1 :

```
-> SV,FL0,fl1
= *** ATTENTION : tous les vol. doivent être PREMARQUES ***
EX-X
= détruire le vol. ** NGF **: ESSAIS (O/N/A=aban.) : O
..... PROGBAL-S      JOURNAL-I      JOURNAL-D      SV-X
RT-X
= SV (RC si fin),
-> -
```

Cet exemple montre la sauvegarde de tous les fichiers de l'unité FL0 sur l'unité FL1. L'utilitaire affiche le nom de chaque fichier à sauvegarder, mais avant de commencer la sauvegarde, il demande aussi la confirmation de la destruction de l'unité de destination. En fin de sauvegarde, il est possible de relancer une nouvelle sauvegarde en complétant la commande. Les nouveaux fichiers à sauvegarder iront alors s'adjoindre à ceux qui ont été précédemment écrits sur l'unité de destination.

Exemple 2 :

```
-> SV,T*-S,FL0
= *** ATTENTION : tous les vol. doivent être PREMARQUES ***
TFP-S
= détruire le vol. ** NGF **: ESSAIS (O/N/A=aban.) : O
..... TTXT-S      TESTBAL-S      TSTFLOP-S      TIMING4-S
TSTSI-S
= SV (RC si fin),B*-S,FL0
= *** ATTENTION : tous les vol. doivent être PREMARQUES ***
BALTX-S      BTX-S      BDCONF-S      BCD-S
=SV (RC si fin),
-> _
```

Dans cet exemple nous avons d'abord la sauvegarde de tous les fichiers dont le nom commence par la lettre "T" et dont le suffixe est "S", puis la sauvegarde de tous les fichiers dont le nom commence par la lettre "B" et dont le suffixe est "S". Tous ces fichiers se trouvent sur le même volume de destination.

Messages d'exécution

Durant la sauvegarde, l'utilitaire affiche le nom du fichier qu'il est en train de sauvegarder. Si un volume n'est pas suffisant pour la sauvegarder, il le signale à l'utilisateur en indiquant le numéro du volume concerné et en demandant de monter le volume sur lequel devra se poursuivre la sauvegarde. Par mesure de sécurité pour vos fichiers, SV vous demandera toujours la confirmation de la destruction du volume de destination. En fin de sauvegarde, la bonne exécution de celle-ci est signalée par l'affichage de "SV (RC si fin)," qui vous permet de relancer une autre sauvegarde sur le volume de destination en cours, ou de revenir à l'interpréteur de commandes par la frappe de <rc>.

Exemple 1 :

```
-> SV,MD0,DEST=FL0,PM
= vol. de réception à PREMARQUER (O/N/A=aband.) : O
= détruire le vol. ** NGF **: COMPTA1 (O/N/A=aband.) : O
..... TEST1-X NSYS-ASM CP-X CPS-X
ASG-X CPM-X DATE-X ED-X EX-X
PROGR001 STATUS-X
= fin de volume numéro:
1 monter le vol. suivant et taper C(A:abandon) : C
= vol. de réception à PREMARQUER (O/N/A=aband.) : O
..... WM-CMD BSC-X PTTY-X TELE-X
DIALOGUE-T DIALOGUE-M PROG002-S
= fin du volume numéro :
2 monter le vol. suivant et taper C(A:abandon) : C
= vol. de réception à PREMARQUER (O/N/A=aband.) : N
= détruire le vol. ** NGF **: PAIE 001 (O/N/A=aband.) : N
= vol. de réception à PREMARQUER (O/N/A=aband.) : N
= détruire le vol. ** NGF **: COMPTA2 (O/N/A=aband.) : O
..... CLIENTS-I CLIENTS-D DEMO-I SFGR-S
SYST00-C DEMOBAL-S
= trop d'extensions dans le fic. : aller sur le vol. suivant
= fin du volume numéro:
3 monter le vol. suivant et taper C(A:abandon) : C
= vol. de réception à PREMARQUER (O/N/A=ABAND.) : N
= détruire le vol. ** COPIE **:
SAUV01 numéro:3 (O/N/A=aband.) : A
-> _
```

L'utilisateur a commencé une sauvegarde de l'unité disque dur MD0 sur l'unité de disques souples FL0. La différence de capacité entre les deux types de périphériques implique qu'il est nécessaire d'avoir plusieurs disques souples. L'option "PM", inscrite à la suite de la commande, indique que l'utilisateur veut avoir la possibilité de prémarquer ses disques souples de l'unité de destination au moment du changement de volume. S'il ne souhaite pas prémarquer son volume, SV lui demandera s'il souhaite détruire le volume qui se trouve sur l'unité.

Toute réponse "A" provoque un abandon de la procédure de sauvegarde et donc un retour à l'interpréteur, et non à la suite de la sauvegarde comme lorsque celle-ci vient de s'achever.

On peut remarquer que le dernier volume à détruire porte, contrairement aux autres, la mention “**COPIE**” et non la mention “**NGF**”. Cette mention est celle des volumes de sauvegarde et indique que ce volume a des fichiers qui ont été constitués par l'utilitaire “SV”.

Messages d'erreurs

Lorsque SV rencontre une erreur de syntaxe ou la référence à un périphérique inconnu, il affiche une erreur numéro 30. D'autres erreurs systèmes ainsi que des messages d'erreur propres à SV peuvent être affichés.

Exemple 1 :

```
—> SV,MD0
      Erreur de Commande n° 30 Syntaxe Incorrecte
—> _
```

Pour toute sauvegarde, l'unité de destination est obligatoire.

Exemple 2 :

```
—> Prologue V2.3 Poste 0 Implicites : FL0,FL0,IM0
—> SV,PROGRAM,FL0
Même support d'origine et de destination
—> _
```

Par cet exemple on remarquera qu'on ne peut sauvegarder un fichier sur le volume où il se trouve. En conséquence, SV ne peut pas servir pour dupliquer directement un fichier sur un même volume.

Exemple 3 :

```
—> SV,FL0,FL1
= *** ATTENTION : tous les vol. doivent être PREMARQUES ***
volume non NGF
—> _
```

SV, signale dans cet exemple que le volume d'origine, bien qu'étant prémarqué et ayant une sectorisation aux normes PROLOGUE, n'a pas une structure NGF.

4.2.7. Restauration des fichiers : "RT"

Présentation

La commande RT permet d'effectuer la restauration intégrale ou partielle d'un volume. Elle ne peut se faire qu'à partir de volumes ****COPIE****, c'est-à-dire de volumes générés par l'utilitaire SV. Lorsque la sauvegarde a été effectuée sur plusieurs volumes, il conviendra, lors de la restauration, de présenter les volumes dans l'ordre dans lequel ils l'ont été lors de cette sauvegarde. On pourra vérifier cet ordre en visualisant le contenu des différents volumes avec l'utilitaire / ; celui-ci reconnaissant les volumes de type ****COPIE****, affiche alors le numéro du volume.

L'utilitaire RT peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (RT est installable en mémoire mais non partageable). Cette installation se fait par la déclaration suivante dans le fichier source.

– RT < INTR

Reportez-vous au chapitre 7 (Configuration des utilitaires résidents) pour plus de détails.

Commande d'appel

—>

[unité1.]RT,nomfich1[[,nomfich2][,nomfich3]...],[DEST=]unité2<rc>

où :

unité1 indique l'unité sur laquelle RT est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire RT est recherché sur l'unité implicite programme de votre poste.

RT nom du fichier binaire correspondant ayant comme suffixe "X".

nomfich désigne le fichier, le groupe de fichiers ou l'unité que vous souhaitez restaurer.

DEST= ce mot désigne la destination et sert donc à différencier l'unité d'origine et l'unité de destination. Ce mot est facultatif dans le cas où la liste ne contient qu'un seul fichier.

unité2 indique l'unité de réception des fichiers à restaurer. Sur cette unité seront montés les différents volumes nécessaires à la restauration.

Remarques :

- La commande ne prévoit pas d'option de prémarquage. Lors d'une restauration totale, il conviendra de prémarquer auparavant le volume de destination.
- Le nom du (ou des) fichier(s) à restaurer ne peut être modifié.

Exemple 1 :

```
-> /,FL0
      Catalogue du Volume Version 3.4b
      Copyright (c) 1983 BULL Micral
*COPIE*
Vol. tstsrt date: (1=dernier) Nbfic:863/6 Lg.Gr:64 Occup:38/20
EX-X      JOURNAL-D      JOURNAL-I      PROGBAL-S      SV-X
RT-X
-> RT,FL0,FL1
EX-X      PROGBAL-S      JOURNAL-I      JOURNAL-D      SV-X
RT-X
-> _
```

Cet exemple montre d'abord le contenu du volume à restaurer. On peut remarquer le message "*COPIE*", qui indique qu'il s'agit d'un volume généré lors d'une sauvegarde, et le message "(1=dernier)", qui signale que ce volume est le premier et le dernier de cette sauvegarde.

Nous avons ensuite, la restauration de tous les fichiers de l'unité FLO sur l'unité FL1. L'utilitaire restaure les fichiers dans l'ordre où ils ont été sauvegardés et affiche au fur et à mesure le nom de chaque fichier. On remarquera que RT ne nous demande pas de monter le volume suivant, la sauvegarde ayant été faite sur un seul volume.

Exemple 2 :

```
-> RT,T-S,FL0  
TFP-S   TTXT-S   TESTBAL-S   TSTFLOP-S   TIMING4-S  
TSTSI-S  
-> _
```

Dans cet exemple, nous avons la restauration de tous les fichiers dont le nom commence par la lettre "T" et dont le suffixe est "S". Là aussi, RT ne demande pas de monter le volume suivant, la sauvegarde ayant été faite sur un seul volume.

Exemple 3 :

```
-> RT,SYSERF-S,SYSERGB-S,SYSCONF-S,DEST=FL0  
SYSERGB-S      SYSERF-S      SYSCONF-S  
-> _
```

Nous avons une restauration sélective de fichiers. Dans cet exemple, le paramètre "DEST=" est obligatoire. L'ordre de restauration des fichiers, n'est pas forcément celui énoncé, mais celui de la sauvegarde.

Messages d'exécution

Durant la restauration, l'utilisateur affiche le nom du fichier qui est en cours de recopie. Si la sauvegarde a été effectuée sur plusieurs volumes, RT demandera le suivant, après avoir exploré le volume déjà monté. Il en contrôlera le nom et le numéro, signalant toute erreur de séquençement afin que l'utilisateur puisse monter le volume adéquat. Une erreur de manipulation n'est donc pas fatale.

Après avoir monté le bon volume, apparaîtra, éventuellement, le message "....." indiquant que l'on continue la restauration d'un fichier déjà commencé avec le précédent volume. La fin de la restauration sera caractérisée par le retour à l'interpréteur de commandes.

Notez qu'à tout moment, c'est-à-dire à chaque message envoyé par RT, l'utilisateur aura la possibilité d'abandonner la restauration en frappant la touche "A".

Lors d'une restauration, RT explore TOUS les volumes de la sauvegarde. Si notre commande contient une liste de fichiers, il peut paraître inutile de monter tous les fichiers les uns après les autres. Il est alors possible d'abandonner la restauration entre deux volumes (en frappant le caractère "A" au moment du changement de volume). Mais attention, le dernier fichier est peut-être à cheval sur le volume en cours et celui à venir ! Abandonner aura alors pour conséquence d'avoir un fichier partiellement restauré et donc inexploitable.

Exemple 1 :

```
-> RT,FL0.FICH0*-* ,FL0.FICH1*-* ,DEST=MD0
FICH01-S   FICH02-S   FICH03-S   FICH04-S   FICH05-S
FICH06-S   FICH07-S
= fin du volume :
1 monter le volume suivant et taper C(A=aband.) : C
.....    FICH08-S   FICH11-S   FICH12-S
-> _
```

Nous avons ici une restauration de plusieurs groupes de fichiers à partir d'une sauvegarde multi-volumes. Après avoir restauré les fichiers du premier volume, RT demande à l'utilisateur de monter le second. Une fois les fichiers de ces volumes restaurés, l'utilitaire reconnaît que le volume monté est le dernier volume de la sauvegarde et retourne à l'interpréteur de commandes.

Messages d'erreurs

Lorsque RT rencontre une erreur de syntaxe ou la référence à un périphérique inconnu, il affiche une erreur système numéro 30. D'autres erreurs systèmes, ainsi que des messages d'erreur propres à RT, peuvent être affichés.

Exemple 1 :

```
-> RT,FL0
      Erreur de commande n° 30 Syntaxe Incorrecte
      Pour toute restauration, l'unité de destination est obligatoire.
-> _
```

Exemple 2 :

-> Prologue V2.3 Poste 0 Implicites : MD0,MD0,IMO

-> RT,PROGRAM,MD0

même support d'origine et de destination

-> _

Exemple 3 :

-> RT,FICH01,FICH02,MD0

DEST= obligatoire

-> _

Lorsqu'une liste de fichiers est énoncée dans la commande,
"DEST" est obligatoire.

Exemple 4 :

-> RT,FL0.FICH01,FL1.FICH02,DEST=MD0

support différent dans 2 sélections

-> _

Si nous voulons restaurer des fichiers provenant de supports
différents, il faudra employer une commande RT par support.

Exemple 5 :

-> RT,FL0,FL1

volume non NGF

-> _

RT signale ici que le volume d'origine, bien qu'étant prémarqué et
ayant une sectorisation aux normes PROLOGUE, n'a pas une
structure NGF.

Exemple 6 :

```
-> RT,FL0.FICH0*.*,FL0.FICH1*.*,DEST=MD0
FICH01-S  FICH02-S  FICH03-S  FICH04-S  FICH05-S
FICH06-S  FICH07-S
= fin du volume:
1 monter le volume suivant et taper C(A=aband.) : C
= err. de seq. des vol., charger la copie:tstsvrt
vol:2 taper C(A:aband.) : C
.....      FICH08-S  FICH11-S
Erreur Périphérique n° 1 Unité Non Prête
= abandon de la restauration reprise sur le volume:2
taper C(A:aband.) : C
.....      FICH08-S  FICH11-S  FICH12-S
-> _
```

Voici les différentes étapes de cet exemple :

- 1 – Lancement de la commande.
- 2 – Restauration des fichiers du premier volume.
- 3 – Fin du premier volume. RT demande de monter le suivant.
- 4 – L'utilisateur monte un autre volume puis frappe <rc> pour continuer la restauration.
- 5 – L'utilitaire refuse le volume monté, et demande que soit monté un volume dont le nom est "tstsvrt" et dont le numéro d'ordre est 2.
- 6 – L'utilisateur monte un autre volume puis frappe <rc> pour continuer la restauration.
- 7 – L'utilitaire accepte le volume, la restauration continue.
- 8 – Une erreur système numéro 1 intervient au cours de la restauration du deuxième volume. RT demande à l'utilisateur s'il faut abandonner ou reprendre la restauration.
- 9 – L'utilisateur frappe "C" pour continuer, c'est-à-dire reprendre la restauration.
- 10 – La restauration reprend alors au DEBUT du volume en cours.
- 11 – La restauration est terminée. Il y a erreur à l'interpréteur de commandes.

4.2.8. **DECONNEXION DU CLAVIER AU PROFIT D'UNE COMMANDE : "ASG"**

Présentation

En cours d'exploitation, il arrive qu'on soit amené à procéder périodiquement, à l'exécution de la même séquence de commandes quelques fois même, avec les mêmes paramètres. Pour éviter de devoir frapper chaque fois ces commandes répétitives au clavier, PROLOGUE vous offre la possibilité de les enregistrer à l'avance dans un fichier de votre choix, et dans l'ordre dans lequel elles devront être exécutées. En utilisant la commande ASG, vous pourrez, au moment opportun, provoquer automatiquement leur exécution en chaîne, sans intervenir au clavier.

Il s'agit donc ici de substituer un fichier de commandes au clavier. Bien entendu, une fois toutes les commandes épuisées, ASG rendra la main à l'interpréteur de commandes en affichant "—>", et le clavier sera alors redevenu disponible.

L'utilitaire ASG peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (ASG est installable en mémoire mais non partageable). Cette installation se fait par la déclaration suivante dans le fichier source "SYSCONF-S" :

— ASG < INTR

Reportez-vous au chapitre 7 (Configuration des utilitaires résidents) pour plus de détails.

Commande d'appel

Première syntaxe

—> [unité.]ASG,desfic[,paramètres] <rc>

où

unité	indique l'unité sur laquelle ASG est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire ASG est recherché sur l'unité implicite programme de votre poste.
ASG	nom du fichier correspondant ayant comme suffixe "X".
desfic	désigne le fichier contenant les commandes explicites ou paramètres. Si le suffixe n'est pas précisé, sa valeur par défaut est "ASG". Mais s'il ne trouve pas de fichier portant ce nom, il prendra "S" comme suffixe.
paramètres	9 au maximum. Ce sont des chaînes de caractères séparées par des virgules destinées à remplacer les paramètres formels contenus dans le fichier source.

Deuxième syntaxe

—> desfic[,paramètres]<rc>

où :

desfic	désigne le fichier source contenant les commandes explicites ou paramètres, le fichier doit avoir obligatoirement "ASG" comme suffixe. Si "ASG" n'est pas résident en mémoire, il faut que IP = IF.
paramètres	9 au maximum. Ce sont des chaînes de caractères séparées par des virgules destinées à remplacer les paramètres formels contenus dans le fichier source.

Il est donc possible d'appeler l'utilitaire "ASG" de deux manières : soit explicitement comme le montre la première syntaxe et dans ce cas n'importe quel fichier source peut lui être associé, soit implicitement comme le montre la deuxième syntaxe et dans ce cas le fichier doit avoir comme suffixe "ASG", ce suffixe ne devant pas être spécifié dans la commande.

On pourra remarquer lors de l'exécution que tous les messages et caractères de sollicitation ne sont pas filtrés, notamment ceux de l'interpréteur de commande ("—>").

Le fichier de commandes

Le fichier peut donc contenir des commandes explicites, c'est-à-dire prises telles qu'elles peuvent être entrées au clavier, mais aussi des paramètres formels ou commandes paramètres. Ces dernières sont en fait des chaînes de caractères qui se composent du caractère "\$" suivi d'un chiffre allant de 1 à 9. Chaque commande paramétrée représente en fait un numéro de paramètre renvoyant au n-ième paramètre situé derrière la commande initiale fournie à l'interpréteur de commandes. Ces paramètres sont séparés par une virgule et sont limités en nombre à 9.

Remarques :

- Dans une commande, un paramètre peut être omis ; il suffira alors d'indiquer son absence par une virgule. Par exemple, dans la commande :

—> ASG,TST1-ASG,PAR1,PAR3,PAR4<rc>

Le deuxième paramètre est inexistant.

- Si dans une commande paramétrée nous utilisons le caractère "\$", il faudra le doubler afin qu'il ne soit pas pris comme caractère d'insertion de paramètre. Ainsi, si nous souhaitons créer un volume dont le nom est \$1, il faudra inscrire dans le fichier source.

CP,CV,FL0,\$\$1

et non

CP,CV,FL0,\$1

- Il est possible d'insérer des lignes de commentaires dans le fichier source. Celles-ci devront commencer par un caractère ";" en début de ligne.

Toute ligne de commentaires sera affichée au terminal.

- Le fichier de commandes est partageable. Il peut donc être utilisé dans un contexte multi-postes, par plusieurs utilisateurs simultanément.
- Une fois lancée, l'exécution de l'utilitaire ASG prendra fin sur la rencontre d'une fin de fichier ou de détection d'une erreur système.

- A tout moment, l'exécution d'ASG peut être suspendue par la frappe de la touche "ESCAPE" ou "SORTIE" <esc> au clavier. L'exécution sera effectivement suspendue lors de la lecture du prochain caractère dans le fichier d'enchaînement.

Deux possibilités se présentent alors :

- Soit la reprise d'ASG par la frappe à nouveau de <esc>
- Soit l'abandon d'ASG par la frappe de <rc>. Dans ce cas, la lecture du fichier d'enchaînement est abandonnée et le clavier est réactivé.

Exemple 1 :

```

-> EDV,TST1-ASG
    Editeur de textes Version x.xx
    Num. Série : # # # # # # # #
    Copyright (c) 1984 BULL Micral
    Nouveau fichier
  
```

Fichier : TST1-ASG _____

```

; Vous êtes dans TST1-ASG
/, *-S
  
```

Fichier : TST1-ASG *E _____

```

-> Prologue V2.3 Poste 0 Implicites : MD0,MD0,IMO
-> ASG,TST1-ASG
-> ; Vous êtes dans "TST1-ASG"
-> /, *-S
    Catalogue du volume Version 3.4b
    Copyright (c) BULL Micral 1983
  
```

Vol: TESTASGdate: 24.9.1984Nbfc :863/297Lg.Gr:64Occup:687/544

```

FICH01-s      FICH02-S
-> ASG,TST1
-> ; vous êtes dans "TST1-ASG"
-> /, *-S
    Catalogue du volume Version 3.4b
    Copyright (c) BULL Micral 1983
  
```

Vol: TESTASG date: 24.9.1984Nbfc:863/297Lg.Gr:64Occup:687/544

FICH01-S

FICH02-S

-> TST1

-> ; Vous êtes dans "TST1-ASG"

-> /,*-S

Catalogue du volume Version 3.4b

Copyright (c) BULL Micral 1983

Vol: TESTASG date: 24.9.1984Nbfc:863/297Lg.Gr:64Occup:687/544

FICH01-S

FICH02-S

-> _

Dans cet exemple, après avoir créé le fichier source qui doit contenir les commandes à exécuter, on montre les trois manières de lancer une exécution d'un fichier source au moyen de l'utilitaire ASG.

La première est la plus explicite. On appelle ASG en le nommant explicitement. Le fichier source, lui aussi explicitement nommé, est alors analysé puis substitué au clavier, caractère par caractère.

La seconde est identique mais ne précise pas le suffixe du fichier source. ASG prend alors par défaut le suffixe "ASG". S'il n'avait pas trouvé de fichier ayant ce nom (ici "TST1-ASG"), il serait allé chercher un fichier avec un suffixe "S" (ici "TST1-S").

La troisième est la plus simple au niveau de l'utilisateur, mais la plus complexe parce que s'analysant au niveau de l'interpréteur de commandes. En effet, celui-ci voyant l'absence de suffixe recherche systématiquement un fichier "X", puis "T", puis "EXE", puis "COM", puis "CMD", enfin "ASG". Celui-ci ayant été trouvé, l'interpréteur de commandes charge alors l'utilitaire ASG-X qui permet de substituer le contenu du fichier "TST1-ASG" au clavier.

Exemple 2 :

-> EDV,TST2-ASG
Editeur de textes Version x.xx
Num. Série : # # # # # # # #
Copyright (c) 1984 BULL Micral
Nouveau fichier

Fichier : TST2-ASG

; Sauvegarde des fichiers “-\$2” (Semaine \$1)
CP,CV,FL0,\$284\$1
O
CP,DF, * -\$2,FL0
/,FL0

Fichier : TST2-ASG * E

-> Prologue V2.3 Poste 0 Implicite : MD0,MD0,IM0
-> TST2,33,DOC
-> ; Sauvegarde des fichiers “-DOC” (semaine 33)
-> CP,CV,FL0,DOC8433 Détruire le volume: XRF0022 (O/N) : O
-> CP,DF,*-DOC,FL/
UTIRT-DOC UTISV-DOC UTIASG-DOC
-> /,fl0
Catalogue du volume Version 3.4b
Copyright (c) BULL Micral 1983

Vol: DOC8433 date: 25.9.1984 Nbfic: 63/13 Lg.Gr:16 Occup:152/94

UTIRT-DOC UTISV-DOC UTIASG-DOC
-> _

Nous montrons ici une procédure de sauvegarde hebdomadaire de groupes de fichiers. Après avoir créé le fichier de commandes “TST2-ASG” au moyen de l’éditeur “EDV”, nous lançons l'utilitaire ASG automatiquement en frappant le nom de notre fichier de commandes suivi de deux paramètres. Le premier représente le numéro de la semaine, le second le suffixe des fichiers à sauvegarder. Au cours de l’exécution, nous pouvons remarquer que “\$1” et “\$2” ont été respectivement remplacés dans les commandes par “33” et “DOC”, y compris dans la ligne de commentaires.

Un fichier de commandes peut comprendre aussi des réponses à des sollicitations de la part d'utilitaires ou de programmes. Ici le "O" (ligne 3 de "TST2-ASG"), correspond à la réponse au message "Détruire le volume" envoyé par l'utilitaire "CP" lors de la création de volume. Lors de sa création, il conviendra donc de bien analyser les commandes du fichier source. Par exemple, ici, nous avons supposé que toute sauvegarde pourra tenir sur un volume et nous n'avons donc pas inséré des réponses sur le multivolume.

Enchaînement de fichiers de commandes

Bien entendu, un fichier de commandes peut contenir une commande ASG. Dans ce cas, il faut considérer celle-ci comme un appel à un sous-programme. Une fois celui-ci exécuté, on reviendra dans le fichier de commande initial, derrière cette commande ASG.

Toutefois, si la commande ASG est la dernière commande du fichier source ET qu'elle est suivie immédiatement d'une fin de fichier, alors l'exécution de cet ASG sera alors un chaînage et non un appel à un sous-programme.

Exemple 1:

```
-> EDV,ASG1-ASG
    Editeur de texte Version x.xx
    Num. Série : # # # # # # # #
    Copyright (c) 1984 BULL Micral
    Nouveau fichier
```

Fichier : ASG1-ASG

```
; Sauvegarde générale (semaine $1)
CP,CV,FLO,GEN84$1
O
ASG2,C,$1
ASG2,I,$1
; Sauvegarde générale terminée
```


Fichier : ASG1—ASG *E

—> EDV,ASG2—ASG

Editeur de textes Version x.xx

Num. Série : # # # # # # # #

Copyright (c) 1984 BULL Micral

Fichier : ASG2—ASG

; Sauvegarde partielle de fichiers “—\$1” (Semaine \$2)

CP,DF,*—\$1,FL0

Fichier : ASG2—ASG *E

—> PROLOGUE V2.3 Poste 1 Implicites : MD0,MD0,IM0

—> ASG1,34 —

—> ; Sauvegarde générale (semaine 34)

—> CP,CV,FL0,GEN8434 Détruire le volume : XRF0025 (O/N) : O

—> ASG2,C,34 —

—> ; Sauvegarde partielle de fichiers “—C” (semaine 34)

—> CP,DF,*—C,FL0

SYST01—C SYST00—C

—> ASG2,I,34 —

—> ; Sauvegarde partielle des fichiers “—I” (semaine 34)

—> CP,DF,*—I,FL0

DOCFICS—I

—> ; Sauvegarde générale terminée

—> —

Sous l'éditeur “EDV” nous sont présentés deux fichiers de commande. Chacun peut être appelé au niveau de l'interpréteur de commandes. Mais le premier fait appel au second pour faire la sauvegarde générale qui est considérée ici comme plusieurs sauvegardes partielles de fichiers. On pourra remarquer que le premier et unique paramètre accolé à la commande au niveau de l'interpréteur de commandes devient le deuxième paramètre dans le fichier de commandes appelé.

Exemple 2 :

-> EDV,BOUCL-ASG
Editeur de texte Version 1.3a
Num. Série : # # # # # # # #
Copyright (c) 1983 BULL Micral
Nouveau fichier

Fichier : BOUCL-ASG

: ; TEST DE BOUCLE
: TEST
: ASG,BOUCL-ASG
:

Fichier : BOUCL-ASG *E

-> BOUCL -
-> ; TEST DE BOUCLE
-> TEST
-> ASG,BOUCL-ASG -
-> ; TEST DE BOUCLE
-> TEST
-> ASG,BOUCL-ASG -
-> ; TEST DE BOUCLE
-> TEST
-> ASG,BOUCL-ASG -
-> ; TEST DE BOUCLE
-> TEST
Erreur de Commande n° 38 Débordement Mémoire
-> -

Dans cet exemple, le fichier de commande se rappelle lui-même. La dernière ligne est terminée par un retour-chariot, ce qui provoque des appels et non des chaînages. De ce fait, "BOUCL—ASG" est chargé en mémoire à chaque appel. La taille de celle-ci n'étant pas illimitée, au bout d'un certain nombre de fois (ici la quatrième) on tombe en débordement mémoire. Cet incident ne se produit pas lorsque la dernière ligne se termine par une fin de fichier, car sur la détection de celle-ci, ASG libère les espaces occupés avant de chaîner sur le nouveau fichier de commandes.

Messages d'exécution et messages d'erreurs

ASG n'envoie aucun message lors de son exécution ni de message d'erreur spécifique. Ceux que nous pouvons rencontrer sont en fait émis par l'exécution des utilitaires ou programmes commandes CONTENUS dans le fichier des commandes.

4.2.9. Gestion des états à imprimer : “SPOOL”

Présentation

Dans une exploitation, les performances générales du système dépendent de nombreux facteurs, parmi lesquels il faut citer en premier la gestion des opérations d'impression. Par nature, une imprimante opère à des vitesses qui sont très lentes par rapport aux vitesses d'accès aux supports sectorisés et à l'exécution de vos programmes. En conséquence, il est souhaitable de mettre en place une méthode de traitement des impressions propre à limiter ces ralentissements. D'autre part, dans un contexte multiposte, la ou les imprimantes sont par nécessité partagées entre les différents postes : un problème d'accès et de disponibilité se pose donc, surtout dans le cas où on ne dispose que d'une seule imprimante.

Enfin, il importe que le matériel d'impression soit utilisé dans les meilleures conditions d'efficacité et de rentabilité. Dans ce dernier cas, c'est l'optimisation de la charge de votre ou vos imprimantes qui doit faire l'objet d'une attention toute particulière.

La commande SPOOL offre un moyen de résoudre ces différents problèmes en vous offrant la possibilité de :

- partager une ressource d'impression entre plusieurs utilisateurs en accès simultané,
- réguler la charge d'une imprimante en vous réservant la possibilité de procéder à des éditions différées,
- optimiser les temps d'exécution des programmes qui comportent un nombre non négligeable de commandes d'impression.

Le principe des opérations consiste à enregistrer, temporairement, sur un support magnétique vos différentes éditions dans une succession de fichiers. Ces textes à imprimer, qui sont appelés “liste”, peuvent être édités parallèlement à l'exécution de vos programmes, une fois votre programme exécuté ou à un moment donné sur ordre de l'utilisateur. Il convient de bien distinguer la phase de mise en fichier et la phase d'impression qui sont désynchronisées mais intimement liées par le fichier “liste” à imprimer.

Les différentes formes de la commande SPOOL vous donnent accès aux fonctions suivantes :

- activation du dispositif de gestion des imprimantes,
- insertion d'un fichier en queue de la file des fichiers à imprimer,
- abandon d'une liste en cours d'impression,
- désactivation du dispositif de gestion des impressions.

Le fichier "liste" :

Le fichier "liste" est donc un fichier créé par le dispositif de gestion des imprimantes (appelé encore communément "SPOULEUR") aussitôt qu'un utilisateur souhaite accéder à l'unité d'impression définie dans la commande d'activation qui a été lancée auparavant sur le même poste ou un poste différent. Un fichier ayant comme nom "//IMp-\$" est créé, où "p" représente le numéro du poste (on remarque immédiatement qu'un poste ne peut créer qu'un seul fichier à la fois). Il reste ouvert jusqu'à ce que l'utilisateur décide de le fermer (en B.A.L., par exemple), ou jusqu'à la fin du programme ayant demandé une impression. Lorsque le fichier est fermé, le "SPOULEUR" le renomme en "//IMn-L" où "n" représente un numéro d'ordre compris entre 1 et 255 inclus. Il reste ainsi nommé jusqu'à son impression totale. Là, il sera SUPPRIME.

L'ensemble des fichiers "//IMn-L" forme une file d'attente des fichiers d'impression. Le premier fichier de la file n'est pas forcément le fichier "//IM1-L", car une partie de la file a pu être imprimée lors de la précédente activation. Le nouveau fichier "liste" est donc mis en bout de file d'attente, derrière celui dont le numéro d'ordre est le plus élevé.

Les fichiers "listes" appartiennent au "SPOULEUR". Il les a créés, il les SUPPRIME. Il convient donc d'être prudent dans la manipulation de ces fichiers, notamment éviter de passer par les utilitaires tels que "CP".

L'utilitaire SPOOL ne peut pas être installé de manière permanente en mémoire. Il vous faut donc l'avoir sur un des supports de votre système. De plus, vous ne pourrez l'utiliser que si les paramètres de protection d'exploitation, spécifiés lors de la génération de votre système, vous y autorisent.

Par contre, l'ensemble SPOULEUR/DESPOULEUR est contenu dans un module extension système, "SPLO1", qu'il est nécessaire de déclater dans SYSCONF-S (Voir chapitre 7).

4.2.9.1. Activation du “SPOUL”

Présentation

L'activation du dispositif de gestion des impressions ne peut se faire que sur une seule unité d'impression (ou imprimante) à la fois. Mais plusieurs postes peuvent adresser cette imprimante. Un fichier “liste” sera créé pour chaque poste.

Pour activer le “SPOULEUR”, l'utilisateur peut choisir entre trois types d'impression :

* L'impression simultanée

Après l'ouverture du fichier “liste”, l'impression de celui-ci, rempli par votre programme, commencera aussitôt que l'imprimante est disponible, après que les fichiers “liste” précédents aient été imprimés dans leur totalité. Si l'imprimante était au repos, donc pas de liste en cours d'édition, l'utilisateur remplit son fichier “liste” pendant que le “SPOULEUR” l'édite sur l'imprimante.

Lorsque l'utilisateur a choisi cette option, le “SPOULEUR” va voir s'il reste des fichiers “liste” (de type “//IMx—L” où x représente le numéro d'ordre) à éditer provenant d'une précédente activation du “SPOULEUR”. Si c'est le cas, il commencera par imprimer celui dont le numéro d'ordre est le plus faible. l'intérêt de ce type d'impression est de pouvoir imprimer immédiatement, si l'imprimante est disponible, par exemple faire une procédure de cadrage ou pouvoir arrêter le programme en fonction de l'impression.

L'édition se fait “modulo 1 secteur”, c'est-à-dire que s'il n'y a qu'un fichier à imprimer le SPOULEUR attendra qu'un secteur soit plein pour l'imprimer ou que ce fichier soit fermé par l'utilisateur.

* L'impression différée

Après son ouverture, le fichier "liste" est rempli par votre programme, mais l'édition ne pourra commencer qu'après la fermeture de ce fichier et que si l'imprimante désignée est disponible. Pendant ce temps de remplissage, le "SPOULEUR" pourra éditer d'autres fichiers "liste" provenant d'une précédente activation.

L'intérêt de ce type d'impression est d'éviter au maximum la dégradation des performances du programme par sa propre édition. Mais cette option n'empêche pas la dégradation des performances par l'édition d'un fichier "Liste" déjà fermé et donc en cours d'édition.

* L'impression inhibée

Après son ouverture, le fichier “liste” est rempli par votre programme, mais aucune édition n’aura lieu même après la fermeture de ce fichier. S’il reste des fichiers “liste”, provenant d’une précédente activation, ils ne seront pas imprimés.

L'intérêt de ce type d'impression est d'éviter au maximum que le "SPOULEUR" n'altère les performances de l'exploitation, notamment dans un contexte multi-postes. Car il faut considérer le "SPOULEUR", en impression, comme un poste aveugle, et plus il y a de postes qui travaillent en même temps, plus les performances de chaque poste sont dégradées. Les impressions sont donc reportées à un autre moment, lorsque l'exploitation sera moins chargée. Lorsque l'utilisateur souhaitera faire ses impressions, il lui suffira d'utiliser une des deux options précédentes.

L'autre intérêt de ce type d'impression est de pouvoir enregistrer toutes ses impressions sur un support amovible et de pouvoir ensuite aller faire les impressions sur un autre système.

Commande d'appel

[illegible]

où :

unité1 indique l'unité sur laquelle l'utilitaire SPOOL est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire SPOUL est recherché sur l'unité implicite programme de votre poste.

SPOOL nom du fichier binaire correspondant ayant comme suffixe "X".

AC commande l'exécution du programme utilitaire SPOOL pour l'opération d'activation du dispositif de gestion des impressions ("SPOULEUR").

DEST= mot-clé lié à la désignation de l'imprimante que l'on veut associer au "SPOULEUR".

nomimp désigne l'imprimante sélectionnée. Si cette option est omise ce sera l'imprimante implicite du poste qui sera prise.

TEMP mot-clé lié à la désignation de l'unité où vont se trouver les fichiers "//IMn-L" créés par le "SPOULEUR".

unité2 désigne l'unité où vont être créés les fichiers à éditer "//IMn-L". Si cette option est omise, l'unité sélectionnée est l'unité implicite système.

NL option : pas de sortie imprimante. Toutes les lignes à imprimer seront insérés dans des fichiers, mais il n'y aura aucune impression.

LDIF option : impression différée. Toutes les lignes à imprimer liées à un programme, seront insérées dans un fichier. L'impression ne pourra commencer que lorsque ce fichier sera fermé (soit par le programme, soit par arrêt du programme) ET lorsque l'imprimante sera disponible.

impression simultanée. Toutes les lignes à imprimer seront insérées dans un fichier. L'impression se fera sitôt l'imprimante disponible. C'est cette option qui sera prise par défaut, si NL, LDIF n'est pas spécifiée dans la commande.

Remarques :

- Quelque soit l'option choisie, tout ordre d'impression provoquera la création d'un fichier "liste". Assurez-vous que vous avez la place nécessaire pour recevoir de tels fichiers sur l'unité choisie.
- L'activation du "SPOULEUR" par un poste est valable pour tout le système et donc pour tous les postes.

Exemple 1 :

- > Prologue V2.3 Poste 1 Implicites : MD0,FL0,IM1
- > SPOOL,AC <rc>

Avec cette commande d'appel, l'utilisateur active le "SPOULEUR". Les fichiers "liste" seront créés sur l'unité implicite système et les impressions se feront sur l'imprimante "IM1" (imprimante implicite du poste 1 sur lequel a été lancé la commande). Le type d'impression est "impression simultanée", c'est-à-dire que l'impression commencera sitôt l'imprimante disponible. Cette commande est équivalente à :

- > SPOOL,AC,DEST=IM1,LISTS<rc>

Exemple 2 :

- > Prologue V2.3 Poste 1 Implicites : MD0,FL0,IM1
- > SPOOL,AC,TEMP=IS,NL <rc>

Avec cette commande d'appel, l'utilisateur active le "SPOULEUR". Les fichiers "liste" seront créés sur l'unité implicite système. Le type d'impression est "impression inhibée" pour l'imprimante "IM1" (implicite imprimante du poste), il n'y aura donc pas de sortie sur cette imprimante. Cette commande est équivalente à :

- > SPOOL,AC,NL <rc>

Messages d'exécution

Après avoir frappé la commande d'appel, l'utilitaire SPOOL analyse la commande et affiche le message "—" indiquant que la commande a été prise en compte et exécutée. A partir de ce moment-là tout programme désirant faire une impression ne recevra plus de messages liés à l'imprimante. C'est donc à l'utilisateur de s'assurer que celle-ci est prête à fonctionner ("sélectionnée", papier disponible, etc...).

Exemple :

```
-> SPOOL,AC -  
-> -
```

Ici, après que l'utilisateur ait frappé la commande d'appel suivi d'un <rc>, l'utilitaire "SPOOL" a activé le "SPOULEUR" puis a affiché le message "—" indiquant que l'activation s'est correctement passée, enfin a rendu la main à l'interpréteur de commande (affichage du message "—>").

Messages d'erreur

Lorsque l'utilitaire SPOOL rencontre une anomalie soit dans la syntaxe de la commande, soit dans l'activation du "SPOULEUR", il le signale à l'utilisateur en lui affichant soit un message système, soit un message qui lui est propre.

* "Fonction Incorrecte"

C'est le message qu'envoie l'utilitaire "SPOOL" lorsqu'il n'a pu reconnaître le type de fonction.

Exemple :

```
-> SPOOL,AX  
Fonction incorrecte  
-> -
```

L'utilisateur a frappé "AX" au lieu de "AC".

* “Paramètre incorrect”

C'est le message qu'envoie l'utilitaire “SPOOL” lorsqu'un mot-clé utilisé est incorrect.

Exemple :

—> SPOOL,AC,DET=IM0

Paramètre incorrect

—> _

Le mot-clé pour désigner l'imprimante de destination est “DEST” et non “DET”.

* “Déjà actif”

C'est le message envoyé par l'utilitaire “SPOOL” lorsqu'on essaie d'activer le “SPOULEUR” alors que celui-ci a déjà été activé, soit sur le même poste soit sur un poste différent.

Exemple :

—> SPOOL,AC —

—> SPOOL,AC

Déjà actif

—> _

La première commande a permis d'activer le “SPOULEUR”, mais le deuxième a été refusée, le “SPOULEUR” ayant été précédemment activé.

* “Système incomplet”

C'est le message envoyé par l'utilitaire “SPOOL” Lorsque celui-ci n'a pu activer le “SPOULEUR” parce que les modules correspondant à celui-ci ne sont pas présents en mémoire. Il convient de vérifier si la mise en place des modules d'extension chargés automatiquement par “SYSCONF—X” lors du chargement du système s'est effectué sans incident et si le fichier source associé à cette procédure, en général “SYSCONF—S”, contient bien le module d'extension “SPLO1” ainsi que le contrôleur imprimante rattaché à “SPLO1”. Pour plus de détails, se reporter au paragraphe 7.4.1. “Configuration des modules d'impression” du chapitre 7.

Exemple :

```
-> SPOOL,AC
Système incomplet
-> EDV,SYSCONF-S
    Editeur de tests Version 1.3a
    Num. Série : # # # # # # # #
    Copyright (c) 1983 BULL Micral
    Ligne à blanc
```

Fichier : SYSCONF-S

```
- SPLO1 < SFX (P1=20H,P2=0)
      < ES 'P1='IM',P2=1,P3='SPLO1')
      < LO
- LOPC< SPLO1
```

Fichier : SYSCONF-S *E

```
-> _
```

La commande d'appel a été refusée par l'utilitaire "SPOOL", le module "SPLO1" n'étant pas présent en mémoire. La visualisation du fichier "SYSCONF-S" au moyen de l'utilitaire "EDV" fait apparaître que le module "SPLO1" ne peut être chargé puisque la ligne où il est défini commence par un ";" signifiant que cette ligne est une ligne de commentaires. Dans ce cas, il faut modifier le fichier "SYSCONF-S" (se reporter au chapitre 7.4 "Configuration des modules d'extension") puis recharger le système.

* "Imprimante inconnue"

C'est le message envoyé par l'utilitaire "SPOOL" lorsqu'on veut déclarer comme imprimante de destination une imprimante qui n'a pas été définie dans le fichier source associé à l'utilitaire "SYSCONF-X" (ce dernier étant exécuté automatiquement lors du chargement du système).

Attention, cette imprimante peut très bien être rattachée physiquement au système, mais pour l'accéder par le "SPOULEUR", elle doit être déclarée obligatoirement dans le fichier source associé à l'utilitaire "SYSCONF-X".

-> SPOOL,AC,DEST=IM2

→ EDV,SYSCONF-S

Num. Série : # # # # # # # # # #

Copyright (c) 1983 BULL Micral

Ligne à blanc

```

:      - SPLO1 < SFX (P1=20H,P2=0)
          < ES (P1='IM',P2=2,P3='SPLO1')
          < LO

```

La commande d'appel "SPOOL" a été refusée, le système ne connaissant pas l'imprimante "IM2". En effet, dans le fichier "SYSCONF-S", visualisé au moyen de l'utilitaire "EDV", on peut remarquer que seulement deux imprimantes ont été déclarées au système, respectivement IM0 et IM1 (paramètre "P2=2"). "IM2" représentant la troisième imprimante, ne peut donc être atteinte.

-> SPOOL,AC,DEST=IP0

Imprimante inconnue

→ EDV,SYSCONF-S

Editeur de textes Version 1.3a

Num. Série : # # # # # # # # # #

Copyright (c) 1983 BULL Micral

Fichier : SYSCONF-S

```
:      - SPLO1 < SFX (P1=20H,P2=0)
              < ES (P1='IM',P2=2,P3='SPLO1)
              < LO
```

Fichier : SYSCONF-S *

La commande d'appel "SPOOL" a été refusée, le système ne connaissant pas l'imprimante "IP0". En effet, dans le fichier "SYSCONF-S", visualisé au moyen de l'utilitaire "EDV", on peut remarquer que le mnémonique utilisé pour désigner une imprimante est "IM" et non "IP".

* "Support disque incorrect"

C'est le système envoyé par l'utilitaire "SPOOL" lorsqu'on veut déclarer comme support pour le fichier "liste" une unité qui n'existe pas physiquement sur le système.

Exemple :

```
-> SPOOL,AC,TEMP=FL1
```

Support disque incorrect

```
-> VPMI
```

Visualisation et modification des implicites de poste Version 1.2a
Copyright (c) 1983 BULL Micral

PROLOGUE V2.3

périphériques : MD FL IM

nombre d'unités : 1 1 2

implicite système : MD0

implicite utilisateur : MD0

implicite imprimante : IM0

Modifications :

implicite système :

L'activation du "SPOULEUR" a été refusée, l'unité "FL1" étant inconnue du système. Ceci se trouve confirmé par l'utilitaire "VPMI" qui affiche une seule unité "FL".

4.2.9.2. Abandon d'une édition en cours

Présentation

Lorsqu'un fichier "liste" est en cours d'impression, il est possible d'interrompre son édition. Le fichier sera alors SUPPRIME et ce sera le fichier "liste" suivant qui sera imprimé, s'il existe.

Commande d'appel

—> [unité1.]SPOOL,AB <rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire SPOOL est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire SPOOL est recherché sur l'unité implicite programme de votre poste.

SPOOL nom du fichier binaire correspondant ayant comme suffixe "X".

AB commande l'exécution du programme utilitaire SPOOL pour l'opération d'abandon de l'impression du fichier "liste" en cours.

Remarque :

— Cette commande ne peut interrompre une édition en mode direct activée par un autre poste.

Messages d'exécution

Après avoir frappé la commande d'appel, l'utilitaire SPOOL analyse la commande puis demande au "SPOULEUR" l'arrêt de l'impression du fichier "liste" en cours, et enfin affiche le message "—" indiquant que la commande a été prise en compte et exécutée.

Exemple :

```
—> SPOOL,AC,LDIF
—> /,MD0,GR,LIS—LO
    Catalogue du volume Version 3.4b
    Copyright (c) 1983 BULL Micral
—> SPOOL,AB —
—> _
```

L'utilisateur a tout d'abord activé le "SPOULEUR" au moyen de la commande "SPOOL,AC". Puis a demandé l'impression du catalogue de l'unité "MD0" sur l'imprimante. Enfin, il a demandé, immédiatement après, l'interruption de l'impression en utilisant la commande "SPOOL,AB".

Messages d'erreur

Lorsque l'utilitaire SPOOL rencontre une anomalie soit dans la syntaxe de la commande, soit dans l'exécution de la commande d'abandon, il le signale à l'utilisateur en lui affichant soit un message système, soit un message qui lui est propre.

• "Non actif"

Ce message indique qu'on souhaite arrêter une impression alors que le "SPOULEUR" n'a pas été activé par une commande "SPOOL,AC".

Exemple :

```
—> SPOOL,AB
Non actif
—> _
```

• "Pas de liste en cours"

Ce message indique que le "SPOULEUR" a bien été activé, mais qu'aucun fichier liste n'est en cours de sortie.

Exemple :

—> SPOOL,AC,NL —

—> SPOOL,AB

Pas de liste en cours

—> _

Après avoir activé le “SPOULEUR” avec une option d'impression inhibée, l'utilisateur a voulu stopper l'impression en cours. Rappelons que la commande “SPOOL” ne peut en aucun cas intervenir sur une impression en directe lancée par un autre poste.

* “Paramètre incorrect”

Ce message apparaît lorsqu'il a été détecté une erreur de syntaxe dans la commande d'appel.

Exemple :

—> SPOOL,AB,NL

Paramètre incorrect

—> _

L'utilisateur ayant plusieurs fichiers “liste” en attente d'impression, après avoir activé le “SPOUL” dans l'intention de les imprimer a décidé de suspendre toutes ses éditions, le fichier en cours d'édition étant perdu. Mais le paramètre “NL” n'existe pas avec la commande “SPOOL,AB”. Noter qu'on ne peut modifier le type d'impression en cours d'édition. Si on souhaite passer d'un type d'impression à un autre, il convient de désactiver le “SPOULEUR” (voir la commande “désactivation du dispositif de gestion des impressions”), puis le réactiver avec le type d'impression désiré.

4.2.9.3. Désactivation du "SPOOL"

Présentation

Une fois le dispositif de gestion des impressions (ou "SPOULEUR") activé, il est possible de le désactiver soit parce qu'on souhaite passer en impression directe, soit parce qu'on souhaite une autre option d'activation. Ainsi un utilisateur ayant mis plusieurs fichiers d'édition en file d'attente, grâce à l'option "NL" dans la commande d'activation, peut après avoir lancé une commande de désactivation lancer son édition au moyen d'une commande d'activation avec une option autre que "NL".

La désactivation du "SPOULEUR" ne sera effective qu'après que le fichier "liste" en cours d'impression, s'il en existe un, ait été imprimé dans sa totalité. Ce fichier "liste" sera alors SUPPRIME, mais les autres de la file d'attente seront laissés tels quels attendant la prochaine activation du "SPOULEUR".

Commande d'appel

—> [unité1.]SPOOL,DS <rc>

unité1 indique l'unité sur laquelle l'utilitaire SPOOL est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire SPOOL est recherché sur l'unité implicite de votre poste.

SPOOL nom du fichier binaire correspondant ayant comme suffixe "X".

DS commande l'exécution du programme utilitaire SPOOL pour l'opération de désactivation du dispositif de gestion des impressions ("SPOULEUR").

Messages d'exécution

Après avoir frappé la commande d'appel, l'utilitaire SPOOL analyse la commande puis demande au "SPOULEUR" de se désactiver. L'affichage du message "—" indique que la commande a été prise en compte et exécutée.

* “Désactivation en cours”

Ce message apparaît lorsqu'un fichier “liste” est en cours d'édition et que la destination ne peut être immédiate.

Exemple 1 :

```
—> SPOOL,AC —  
—> SPOOL,AB —  
—> SPOOL,DS      Désactivation en cours —  
—> _
```

La première commande “SPOOL” avec la fonction “AC” a permis d'activer le “SPOULEUR”. L'utilisateur ayant décidé d'interrompre toutes ses éditions, frappe une deuxième commande “SPOOL” avec la fonction “AB” pour stopper l'édition en cours, enfin frappe une troisième commande “SPOOL” avec la fonction “DS” pour désactiver le “SPOULEUR”. Mais attention, cet enchaînement de commandes non seulement a SUPPRIME le premier fichier “liste”, mais aussi le second ! En effet, la commande “SPOOL,AB” a provoqué l'abandon de l'impression du fichier “liste” en cours et a immédiatement lancé l'impression du fichier “liste” suivant.

L'impression de ce dernier aura lieu dans sa totalité et la désactivation se fera une fois l'impression du second fichier “liste” terminée. Les autres fichiers restent intacts.

Exemple 2 :

```
—> SPOOL,AC —  
—> SPOOL,DS      Désactivation en cours —  
—> SPOOL,AB —  
—> _
```

La première commande "SPOOL,AC" a permis d'activer le "SPOULEUR". L'utilisateur ayant décidé d'interrompre toutes ses éditions, frappe une commande "SPOOL,DS" pour désactiver le "SPOULEUR". Mais celui-ci est en cours d'édition (d'où message "Désactivation en cours"), la désactivation ne deviendra effective qu'après l'édition du fichier "liste" en cours. L'utilisateur enfin frappe une commande "SPOOL,AB" pour abandonner l'impression en cours. Cette commande n'est pas rejetée car le "SPOULEUR" est toujours considéré comme actif. L'édition sera stoppée immédiatement, le fichier "liste" qui était en cours d'impression est SUPPRIME et les suivants sont intacts. Contrairement à l'exemple précédent, où l'arrêt n'avait pu se faire sur le fichier en cours alors que nous souhaitons l'arrêt immédiat de l'impression des fichiers "liste", nous nous sommes arrêtés sur le fichier "liste" voulu et lui seul a été détruit.

Exemple 3 :

```

-> SPOOL,AC,LDIF-
-> /,GR,LIS-LO
    Catalogue du volume Version 3.4a
    Copyright (c) 1983 BULL Micral
-> SPOOL,DS      Désactivation en cours
-> CP,DF,SYSCONF-S,LIS-LO
    Erreur Fichier n° 44 Fichier Non Partageable
-> _

```

La première commande "SPOOL,AC" active le "SPOULEUR" avec l'option d'impression différée. Puis l'utilisateur demande le contenu du catalogue du support se trouvant sur l'unité implicite utilisateur. L'édition commence une fois le fichier "liste" fermé, c'est-à-dire entièrement rempli. L'utilisateur décide de désactiver le "SPOULEUR" pour passer en impression directe. La commande suivante, de copie d'un fichier source sur l'imprimante, est refusée car l'impression via le "SPOULEUR" n'est pas terminée et l'imprimante ne peut être accédée immédiatement en même temps que le "SPOULEUR".

Messages d'erreur

Lorsque l'utilitaire SPOOL rencontre une anomalie, soit dans la syntaxe de la commande, soit dans l'exécution de la commande d'abandon, il le signale à l'utilisateur en lui affichant, soit un message système, soit un message qui lui est propre.

* "Paramètre incorrect"

Ce message est envoyé si la commande "SPOOL,DS" est suivie d'un paramètre.

Exemple :

—> SPOOL,DS,TEMP=MD0

Paramètre incorrect

—

"TEMP=MD0" est inutile. Rappelons que le "SPOULEUR" ne gère qu'une seule imprimante, et que les fichiers "liste" associés sont tous localisés sur une même unité déclarée au moment de l'activation.

* "Non actif"

Ce message est envoyé si on tente de désactiver le "SPOULEUR", alors qu'il n'a jamais été activé, ou qu'il a déjà été désactivé.

Exemple :

—> SPOOL,AC,NL —

—> SPOOL,DS —

—> SPOOL,DS

Non actif

—> —

La dernière commande "SPOOL,DS" sera refusée, le "SPOULEUR" ayant été désactivé par la précédente. Rappelons que cette commande n'a d'effet sur aucune imprimante en accès direct.

4.2.9.4. Insertion d'un fichier pour édition

Présentation

Il est possible d'ajouter en queue de la file d'attente des fichiers "liste" à imprimer, un fichier de type source se trouvant sur le même support que ceux-ci. Le fichier sera alors renommé en "//IMn-L" où n représente son rang dans la file d'attente. Une fois que le fichier aura été imprimé par le "SPOULEUR", il sera SUPPRIME du support comme tous les autres fichiers "liste".

Attention, si le fichier de type source avait des clés d'accès, celles-ci ne seront pas transmises au nouveau nom "//IMn-L".

Commande d'appel

—> [unité1.]SPOOL,IL,desfic <rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire SPOOL est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire SPOOL est recherché sur l'unité implicite programme de votre poste.

SPOOL nom du fichier binaire correspondant ayant pour suffixe "X".

IL commande l'exécution du programme utilitaire SPOOL pour l'opération d'insertion d'un fichier de type source dans la file d'attente des fichiers "liste".

Exemple :

—> SPOOL,AC,TEMP=FL0,DEST=IM1,LDIF —

—> SPOOL,IL FL0.FICH-S <rc>

Avec la commande "SPOOL,IL", l'utilisateur souhaite insérer le fichier "FICH-S" à la suite des fichiers "liste" déjà existants —en cours d'impression sur l'imprimante "IM1"— sur le même support, c'est-à-dire "FL0". Avant de le mettre en file d'attente le "SPOULEUR" le renommiera. Après impression, FICH-S sera détruit.

Messages d'exécution

Après avoir frappé la commande d'appel, l'utilitaire SPOOL analyse la commande puis insère le fichier spécifié dans la file d'attente. L'affichage du message "—" indique que la commande a été prise en compte et exécutée.

Exemple :

—> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0

—> /,/IM*—L

Catalogue du volume Version 3.4.b

Copyright (c) 1983 BULL Micral

Vol: voltest date: 14.9.1984 Nbfic: 863/288 Lg.Gr: 64 Occup: 687/552

Erreur fichier n° 40 fichier inexistant

—> SPOOL,AC,NL —

—> SPOOL, IL,FICH01—S

—> /,/IM*—L

Catalogue du volume Version 3.4b

Copyright (c) 1983 BULL Micral

Vol: voltest date: 14.9.1984 Nbfic: 863/288 Lg.Gr: 64 Occup: 867/552

//IM1—L

—> /,FICH01—S

Catalogue du volume Version 3.4b

Copyright (c) 1983 BULL Micral

Vol: voltest date: 14.9.1984 Nbfic: 863/290 Lg.Gr: 64 Occup: 687/554

Erreur fichier n° 40 fichier inexistant

—> —

L'utilisateur, par la commande d'appel "SPOOL,IL" a rajouté le fichier "FICH01—S" à la file d'attente des fichiers "liste". Avec les commandes "/", on peut remarquer que ce fichier est devenu "//IM1—L" et qu'il n'existe plus sous son nom d'origine. Lorsque le "SPOULEUR" sera activé avec une autre option que "NL", "//IM1—L" sera imprimé, puis SUPPRIME du catalogue.

Messages d'erreur

Lorsque l'utilitaire SPOOL rencontre une anomalie soit dans la syntaxe de la commande, soit dans l'exécution de la commande d'abandon, il le signale à l'utilisateur en lui affichant soit un message système, soit un message qui lui est propre.

* "Paramètre incorrect"

Ce message apparaît lorsqu'il a été détecté une erreur de syntaxe dans la commande d'appel.

Exemple :

—> SPOOL,IL FICH01—S,NL

Paramètre incorrect

—> _

Le paramètre "NL" n'existe pas dans la syntaxe de la commande d'appel "SPOOL,IL". On notera qu'avec l'insertion d'un fichier, il est impossible de modifier le type d'impression. On ne peut avoir, par exemple, une partie des fichiers "liste" qui sont en sortie différée et une autre sans impression : le type d'impression est lié à toute la file d'attente et non à un groupe de fichiers.

* "Non actif"

Ce message apparaît lorsqu'on essaie d'insérer un fichier dans la file d'attente alors que le "SPOULEUR" n'est pas actif. L'insertion d'un fichier dans une file d'attente ne se limite pas à renommer ce fichier mais à mettre à jour des pointeurs début de file, fin de file, pointeur courant... C'est pour cela que la commande "SPOOL,IL" nécessite un "SPOULEUR" actif, même si l'option d'impression est "NL".

Exemple :

```
-> SPOOL,DS Désactivation en cours
-> SPOOL,IL,FICH01-S -
-> SPOOL,IL FICH02-S
Non actif
-> _
```

Après avoir demandé la désactivation du "SPOULEUR", l'utilisateur a pu insérer un fichier dans la liste d'attente car la désactivation n'était pas encore effective (lors de la commande de désactivation, un fichier était en cours de sortie et la désactivation ne se fait que lorsque le fichier est totalement imprimé et SUPPRIME). Par contre, la deuxième insertion dans la file d'attente (troisième commande d'appel SPOOL) a été refusée, la désactivation étant intervenue entre temps.

* "Support incompatible"

Ce message apparaît lorsque le fichier à insérer dans la file d'attente ne se trouve pas sur le même support que les fichiers "liste".

Exemple :

```
-> SPOOL,AC,TEMP=FL0,NL -
-> SPOOL,IL,MD0.FICH01-S
Support incompatible
-> _
```

La commande d'activation du "SPOULEUR" indique que les fichiers "liste" à traiter vont se trouver sur le support "FL0". La commande d'insertion du fichier "FICH01-S" dans la file d'attente est refusée, car ce fichier se trouve sur le support "MD0". Pour résoudre ce problème, il faut utiliser la commande "CP" comme suit :

```
-> SPOOL,AC,TEMP=FL0,NL -
-> CP,DF,MD0.FICH01-S,LIS=LO -
-> CP,SF,MD0.FICH01-S -
-> _
```


La commande "CP,DF" recopie le fichier sur le support implicite imprimante. Mais comme le "SPOULEUR" a été activé sur celle-ci, le fichier sera recopié sur le support contenant les fichiers "liste" ("FL0"). Bien sûr le fichier est renommé sous la forme "//IMn-L".

Enfin pour avoir une fonctionnalité rigoureusement identique à la commande "SPOOL,IL", il convient de supprimer le fichier d'origine via la commande "CP,SF". Ainsi, lorsque l'impression sera faite, il n'y aura plus aucune trace du fichier à imprimer, comme avec la commande "SPOOL,IL".

4.2.9.5. Utilisation du "SPOOL" en multiposte

Le tableau suivant est destiné à montrer les différentes phases et leur imbrication pour deux postes et le "SPOULEUR". Chacun des postes demande la copie du contenu d'un fichier source sur imprimante, via le "SPOULEUR".

Chrono	Poste 0	Poste 1	Spouleur
T0 T1 T3	-> SPOOL,AC,LDIF - -> CP,DF,F1-S,LIS=LO		Création et ouv. fichier // IM0-\$
T4 T5		-> CP,DF,F2-S,LIS=LO	Création et ouv. fichier //IM1-\$
T6 T7		-> -	Fin poste 1, donc fermeture //IM1-\$ Renommé en //IM1-L. Début impresion
T8 T9	-> -		Fin poste 0, donc fermeture //IM0-\$ Renommé en //IM2-L. Fin d'impression de //IM1-L.
T10			Suppression de //IM1-L. Début d'impression de //IM2-L.
T11			Fin d'impression de //IM2-L. Suppression de //IM2-L. Attente d'un prochain fichier.

Remarques :

La traitement des deux commandes "CP" s'adressant à une seule imprimante se fait en parallèle. Dans un contexte sans activation du "SPOULEUR", le poste 1 n'aurait pu lancer sa commande qu'après l'édition totale résultant de la commande du poste 0.

Les deux postes restent bloqués le temps de la recopie de leur fichier à éditer dans un fichier "liste".

Le poste 1, bien qu'ayant lancé sa commande après le poste 0, est libéré le premier (le fichier F2—S doit être de taille moins importante que le fichier F1—S).

4.2.10. Emulation d'un volume en mémoire : "MEMDISK"

Présentation

La commande MEMDISK permet l'émulation d'un volume PROLOGUE en mémoire. Son interface d'appel est identique à celui d'un contrôleur disque. Cependant, on ne peut pas le prémarquer. Pour une utilisation efficace, ce module nécessite 256 Ko d'espace mémoire.

Ses caractéristiques sont :

- mnémonique : RM
- nb d'unité : 1
- nb de secteurs / piste : 16
- taille piste : 4 K

Avant d'utiliser la mémoire virtuelle autrement qu'en accès direct, il faut faire une création de volume de la ressource mémoire, sans option PM :

Exemple :

—> CP,CV,RM0,NOMMEM

puis faire éventuellement

VPMI avec implicite fichier = RM0

Exemple :

—> /,RM0<rc>

Cette commande affichera sur l'écran le contenu de la mémoire virtuelle RM0 que vous avez créé.

—> /,RM0

Catalogue du volume Version 3.4d

Copyright (c) 1983 BULL Micral

Vol : NOMMEM Date : 1.1.1985 Nbfic : 63/1 Lg.Gr : 16 Occup : 24/3

TOTO—S

4.3. LES COMMANDES A USAGE SYSTEME

4.3.1. Visualisation et mise à jour des secteurs disque : "PATCH"

Présentation

PATCH est un utilitaire qui permet de visualiser, et éventuellement modifier, le contenu d'un secteur d'un volume sectorisé (disque dur, disque souple). Ce secteur peut être atteint soit directement en spécifiant son adresse sur le disque, soit indirectement en spécifiant un numéro de secteur dans un fichier.

Les modifications qui peuvent être effectuées sont en fait des remplacements de codes ASCII ou hexa par d'autres codes ASCII ou hexa et non des insertions.

Cet utilitaire est à manier avec prudence, car une modification peut avoir des conséquences facheuses sur l'exploitation. Par exemple, si nous remplaçons un caractère "fin de fichier" dans un fichier de type "source" par n'importe quoi, lors de la reprise de fichier avec "EDV", nous aurons l'affichage de lignes que nous n'avons pas créées et non significatives.

L'utilitaire PATCH ne peut pas être installé de manière permanente en mémoire. Il doit donc se trouver sur un des supports de votre système. De plus, vous ne pourrez l'utiliser que si les paramètres de protection d'exploitation, spécifiés lors de la génération de votre système, vous y autorisent. Pour plus de précision sur ces problèmes de protection d'exploitation et leur mise en œuvre, reportez-vous au chapitre 7 (Configuration de PROLOGUE).

Commandes d'appel

Première syntaxe :

—> [unité1.]PATCH,unité2[,NPIS=np][,NSEC=ns]<rc>

où :

- unité1 indique l'unité sur laquelle l'utilitaire PATCH est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire PATCH est recherché sur l'unité implicite programme de votre poste.
- PATCH nom du fichier binaire correspondant ayant comme suffixe "X".
- unité2 unité sur laquelle se trouve le ou les secteur(s) que nous souhaitons visualiser et éventuellement modifier.
- NPIS= mot-clé indiquant que l'on veut spécifier un numéro de piste. Par défaut, c'est zéro qui est pris.
- np numéro de la piste dans laquelle se trouve le premier secteur que l'on veut visualiser. Ce numéro peut être exprimé en hexadécimal (nh).
- NSEC= mot-clé indiquant que l'on veut spécifier un numéro de secteur. Par défaut, c'est zéro qui est pris.
- ns numéro du premier secteur que l'on désire visualiser. Si le paramètre "NPIS=" est spécifié, ce numéro de secteur sera relatif au début de la piste. S'il n'est pas spécifié, ce sera un numéro de secteur absolu (c'est-à-dire par rapport au début du disque).

Deuxième syntaxe :

—> [unité1.]PATCH,desfic[,NSEC=ns]<rc>

où :

- unité1 indique l'unité sur laquelle l'utilitaire PATCH est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire PATCH est recherché sur l'unité implicite programme de votre poste.
- PATCH nom du fichier binaire correspondant ayant comme suffixe "X".
- desfic fichier dont nous souhaitons visualiser et éventuellement modifier un ou plusieurs secteurs.
- NSEC= mot-clé indiquant que l'on veut spécifier un numéro de secteur. Par défaut, c'est zéro qui est pris.
- ns numéro du premier secteur que l'on désire visualiser. C'est un numéro de secteur relatif au début de fichier.

Remarques :

- Tout numéro de piste ou numéro de secteur est compté à partir de zéro.
- Les valeurs np et ns sont implicitement données en décimal. Elles peuvent être données en hexadécimal si elles sont suivies du caractère "H" (ou "h").
- Avec la première syntaxe, il est possible d'atteindre n'importe quelle zone utilisable du disque même celles non encore utilisées.
- Lorsqu'on utilise la première syntaxe, le numéro de secteur s'il est spécifié relativement à une piste, devra être inférieur au nombre de secteurs par piste pour le support désiré.
- Lorsqu'on utilise la deuxième syntaxe avec un numéro de secteur, celui-ci doit être inférieur au nombre total de secteurs constituant le fichier. Car avec cette syntaxe, on ne peut déborder du fichier nommé.

Exemples :

—> PATCH,FL0 <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 0 (premier secteur) du disque souple qui se trouve sur l'unité FL0.

—> PATCH,FL0,NSEC=100 <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 100 du disque souple qui se trouve sur l'unité "FL0".

—> PATCH,FL0,NPIS=7 <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 0 de la piste 7 du disque souple qui se trouve sur l'unité "FL0".

—> PATCH,FL0,NPIS=7,NSEC=14 <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 14 de la piste 7 du disque souple qui se trouve sur l'unité FL0.

—> PATCH,FICTEST <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 0 (premier secteur) du fichier "FICTEST".

—> PATCH,FICTEST,NSEC=5 <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 5 (sixième secteur) du fichier "FICTEST".

—> PATCH,FICTEST,NSEC=12 h <rc>

Avec cette commande, l'utilisateur va visualiser le secteur 18 (décimal, = 12 hexadécimal) du fichier "FICTEST".

La visualisation d'un secteur

Une fois votre commande frappée, l'utilitaire PATCH efface votre écran puis vous affiche une série d'informations qui sont réparties en quatre zones :

ZONE 1

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF		
09	09	09	09	50	20	41	20	54	20	43	20	48	0D	0A	09	00	... P A T C H ...	00
09	09	09	2D	2D	2D	2D	2D	2D	2D	2D	2D	0D	0A	0D	0A	10		10
43	65	20	66	69	63	68	69	65	72	20	65	73	74	20	64	20	Ce fichier est d	20
65	73	74	69	6E	65	20	61	20	65	74	72	65	20	75	74	30	estiné à être ut	30
69	6C	69	73	65	20	70	61	72	20	6C	27	75	74	69	6C	40	ilisé par l'util	40
69	73	61	74	65	75	72	20	70	6F	76	72	20	76	6F	69	50	isateur pour voi	50
72	20	6C	65	73	0D	0A	64	69	66	66	65	72	65	6E	74	60	r les -different	60
65	73	20	66							6E	6E	61	6C	69	74	70		70
																80	ZONE	80
																90		90
72	65	20	22							2E	20	49	6C	20	63	A0	onvient d'être	A0
6F	6E	76	69	65	6E	74	20	64	27	65	74	72	65	0D	0A	B0	prudent dans la	B0
70	72	75	64	65	6E	74	20	64	61	6E	73	20	6C	61	20	C0	manipulation des	C0
6D	61	6E	69	70	75	6C	61	74	69	6F	6E	20	64	65	73	D0	commandes afin	D0
20	63	6F	6D	6D	61	6E	64	65	73	20	61	66	69	6E	20	E0	de ne pas casser	E0
64	65	20	6E	65	20	70	61	73	20	63	61	73	73	65	72	F0	le texte -autre	F0
20	6C	65	20	74	65	78	74	65	0D	0A	61	75	74	72	65			

ZONE 4

- La PREMIERE zone est constituée d'informations situant le secteur qui est affiché. Nous y trouvons le nom du support si nous avons utilisé la première syntaxe, le nom du fichier, si nous avons utilisé la deuxième ainsi que le numéro du secteur qui est affiché.
- La DEUXIEME zone est composée de nombre codés en hexadécimal et qui représentent les 256 octets du secteur défini dans la ZONE 1. Cette zone est surmontée d'une ligne contenant des nombres (00 01 02 03 04 ...) : ils représentent les poids faibles (en hexadécimal) de la numérotation des octets. A droite de cette zone, nous trouvons aussi des nombres (00 10 20 30 ...) : c'est la numérotation des octets qui sont situés sur la même ligne et à gauche dans la zone.

— La TROISIEME zone est composée de caractères. C'est la visualisation des codes ASCII qui constituent le secteur. Pour chaque ligne, chaque caractère a son correspondant codé en hexadécimal dans la ZONE 2. On peut remarquer que tous les codes hexadécimaux ne sont pas visualisés par un caractère. Seuls les caractères dont la valeur est comprise entre 20 hexa et 7F hexa inclus, sont visualisés. Les autres sont visualisés par des points.

A droite de notre zone, nous trouvons aussi des nombres (00 10 20 30 ...) : C'est la numérotation des octets qui sont situés sur la même ligne et à gauche dans notre zone (poids forts).

— La QUATRIEME zone est réservée aux messages qu'affiche PATCH et aux paramètres que l'utilisateur doit rentrer avec ces commandes.

Le déplacement du curseur et modification d'un caractère

Une fois que notre commande lancée à l'interpréteur a été exécutée, nos quatre zones sont remplies. On peut remarquer la présence du curseur sur le premier caractère de la visualisation hexadécimale du premier octet (octet 0) du secteur dans la ZONE 2. Le caractère équivalent codé en ASCII et se trouvant dans la ZONE 3 est mis en valeur par un attribut de visualisation.

Nous pouvons à ce moment là utiliser les touches de déplacement du curseur, pour atteindre n'importe quel caractère hexadécimal du secteur visualisé. Parallèlement, on visualise l'équivalent codé ASCII avec l'attribut préférentiel. Si nous souhaitons revenir dans le coin en haut à gauche de la ZONE 2 frapper la touche "home" (ou CTRL—□) suffit.

Lorsque le caractère hexadécimal désiré est atteint, il est possible de modifier celui-ci en frappant le code hexadécimal désiré. Immédiatement, apparaît dans la ZONE 3 le nouveau caractère correspondant. Cette modification n'est pas enregistrée dans le fichier. Ceci peut être répété pour autant de caractères que l'on désire dans le secteur.

Les caractères hexadécimaux possibles sont :

- "0" à "9"
- "A" à "F" ou "a" à "f"

-> PATCH,FL0.F-S <rc>

Fichier : FLO.F S																Secteur : 0																
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF																
09	09	09	09	50	20	41	20	54	20	43	20	48	0D	0A	09	00	--- P A T C H ---															00
09	09	09	2D	2D	20	41	2D	2D	2D	2D	2D	0D	0A	0D	0A	10																10
43	65	20	66	69	63	68	69	65	72	20	65	73	74	20	64	20	Ce fichier est d															20
65	73	74	69	6E	65	20	61	20	65	74	72	65	20	75	74	30	estimé à être ut															30
69	6C	69	73	65	20	70	61	72	20	6C	27	75	74	69	6C	40	ilisé par l'utili															40
69	73	61	74	65	75	72	20	70	6F	76	72	70	6E	6F	69	50	isateur pour voi															50
72	20	6C	65	73	0D	0A	64	69	66	66	65	72	65	6E	74	60	r les -différent															60
65	73	20	66	6F	6E	63	74	69	6F	6E	6E	61	6C	69	74	70	es fonctionnalit															70
65	73	20	64	65	20	6C	27	75	74	69	6C	69	74	61	69	80	es de l'utilitai															80
72	65	20	22	50	41	54	53	48	22	2E	20	49	6C	20	63	90	re «PATCH». Il c															90
6F	6E	76	69	65	6E	74	20	64	27	65	74	72	65	0D	0A	A0	onvient d'être															A0
70	72	75	64	65	6E	74	20	64	61	6E	73	20	6C	61	20	B0	prudent dans la															B0
6D	61	6E	69	70	75	6C	61	74	69	6F	6E	20	64	65	73	C0	manipulation des															C0
20	63	6F	6D	6D	61	6E	64	65	73	20	61	66	69	6E	20	D0	commandes afin															D0
64	65	20	6E	65	20	70	61	73	20	63	61	73	73	65	72	E0	de ne pas casser															E0
20	6C	65	20	74	65	78	74	65	0D	0A	61	75	74	72	65	F0	le texte -autre															F0



* H (ou touche CTRL-J) : Mémento des commandes de PATCH

Cette commande permet d'afficher un mémento des différentes commandes de PATCH dans la ZONE 2. De plus sont affichées dans la ZONE 4 les caractéristiques du support, si nous avons utilisé la première syntaxe et les caractéristiques du fichier, si nous avons utilisé la deuxième syntaxe.

Une fois ce mémento affiché, frapper n'importe quelle touche permet de réafficher le secteur tel qu'il était visualisé avant la commande "H". En aucun cas la commande "H" n'affecte le support.

Exemple :

Après une commande "H", nous avons :

Fichier : FLO.F5																Secteur : 0																	
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF																	
Menu																00	---P A T C H---																00
: Secteur Suivant																10	-----																10
P : Secteur Précédent																20	Ce fichier est d																20
L : Lire un Secteur ou Piste,Secteur																30	estiné à être ut																30
M : Ecrire le Secteur en cours																40	ilisé par l'util																40
T : Translater le Secteur en cours																50	isateur pour voi																50
K : Rechercher une Chaîne																60	r les--different																60
N : Nouvelle Recherche																70	es fonctionnalit																70
I : Imprimer le Secteur en cours																80	es de l'utilitai																80
S : Impression enchainée																90	re «PATCH». Il c																90
R : Visualisation enchainée																A0	onvient d'être--																A0
Z : Remplissage du secteur avec une valeur																B0	prudent dans la																B0
X : Convertir une chaîne en hexadécimal																C0	manipulation des																C0
home : Curseur en haut à gauche																D0	commandes afin																D0
Esc : Sortir																E0	de ne pas casser																E0
Frappier une touche pour retour																F0	le texte--autre																F0

Caractéristiques : 16 Secteurs

On peut remarquer que le contenu de la ZONE 3 est la visualisation d'un secteur dont la définition apparaît dans la ZONE 1. Dans la ZONE 4 apparaît des caractéristiques du fichier : une taille de 16 secteurs.

''': Secteur Suivant

La frappe de la barre d'espacement (20 hexa) provoque l'affichage du secteur suivant. Les quatre zones sont mises à jour et le curseur se place dans le coin en haut à gauche de la ZONE 2. Si le secteur suivant n'existe pas (fin du volume pour la première syntaxe ou fin de fichier pour la deuxième), un signal sonore est envoyé et les quatre zones restent inchangées.

*** P : Secteur Précédent**

La frappe de ce caractère provoque l'affichage du secteur précédent. Les quatre zones sont mises à jour et le curseur se place dans le coin en haut à gauche de la ZONE 2. Si le secteur précédent n'existe pas (début de volume pour la première syntaxe ou début de fichier pour la deuxième), un signal sonore est envoyé et les quatre zones restent inchangées.

*** L : Lire un Secteur ou Piste, Secteur**

En plus de la lecture du secteur précédent ou du secteur suivant, PATCH permet d'aller lire un secteur donné. Après le message demandant ce numéro de secteur, l'utilisateur peut frapper directement un numéro de secteur (suivi d'un <rc>) qui sera absolu si la commande d'appel a la première syntaxe et sera relatif au début de fichier avec la deuxième syntaxe. Dans le cas de la première syntaxe, l'utilisateur peut frapper une réponse du type "p,s" (suivi d'un <rc>) où "p" représente un numéro de piste et "s" un numéro de secteur dans cette piste. Si le secteur ne peut être lu, un signal sonore est émis, les quatre zones restent inchangées, PATCH redemande un numéro de secteur. La frappe d'un <rc> seul provoque l'abandon de cette commande "L". La ZONE 4 sera alors effacée, les autres inchangées et le curseur reviendra se positionner à la place où il se trouvait avant la frappe de la commande "L".

Une fois le secteur lu, les quatre zones sont mises à jour et le curseur se positionne dans le coin gauche de la ZONE 2.

*** M : Ecrire le Secteur en cours**

C'est la commande LA PLUS DANGEREUSE DE PATCH. Il convient donc de la manier avec prudence.

Lorsqu'après avoir modifié un ou plusieurs caractères sur le secteur en cours (secteur visualisé), on désire modifier ce secteur sur le support, il convient de frapper la commande "M". Le message suivant apparaît alors : "M : Ecrire le Secteur en cours". L'utilisateur attend une confirmation de la part de l'utilisateur. La frappe d'un <rc> provoquera la réécriture du secteur. Puis la ZONE 4 sera effacée et le curseur viendra se placer dans le coin en haut à gauche de la ZONE 2. Si après le message "M : Ecrire le Secteur en cours", l'utilisateur décide de renoncer à la réécriture du secteur en cours, la frappe d'une touche autre que <rc> effacera la ZONE 4 et le curseur reviendra se positionner là où il était avant la frappe de la commande "M".

*** T : Translater le Secteur en cours**

Cette commande permet de recopier le contenu du secteur en cours, tel qu'il est affiché, dans un autre secteur. Après la frappe de la commande, PATCH affiche le message "Ecrire Secteur N° :" dans la ZONE 4 et attend le numéro du secteur de destination. L'utilisateur a le choix entre abandonner en frappant la touche <rc>, dans ce cas le curseur revient à sa position initiale et la ZONE 4 est effacée et frapper le numéro du secteur de destination suivi d'un <rc>. Dans ce cas, le secteur visualisé sera immédiatement transféré dans le secteur spécifié. La ZONE 4 sera ensuite effacée et le curseur se positionnera dans le coin en haut à gauche de la ZONE 2, les autres zones restant inchangées.

Si le secteur de destination ne peut être atteint, PATCH émet un signal sonore et réaffiche le message "Ecrire Secteur N° :" dans la ZONE 4 en attendant une nouvelle réponse.

• K : Rechercher une chaîne

Cette commande permet de rechercher une chaîne de caractères à partir de la position courante du curseur dans le secteur visualisé. La recherche qui ne se borne pas au secteur visualisé, peut s'effectuer dans le sens croissant ou décroissant au choix de l'utilisateur. Elle s'arrête lorsque la chaîne est trouvée ou lorsqu'on atteint un début ou une fin de fichier (première syntaxe), ou lorsqu'on atteint un début ou une fin de volume (deuxième syntaxe).

Après avoir frappé la commande "K", apparaît —dans la ZONE 4— le message "Chaîne :". L'utilisateur doit alors rentrer la chaîne qu'il désire rechercher suivie d'un <rc>. Cette chaîne peut être hexadécimale, dans ce cas elle doit être précédée du caractère "/", ou peut être une chaîne ASCII (cette chaîne ne peut commencer par "/"). Une fois la chaîne entrée, PATCH affiche le message "Sens de la Recherche (Croissant, Décroissant) : C". L'utilisateur frappera alors <rc> s'il veut l'ordre croissant et "flèche gauche" suivi de "D" et <rc> s'il veut l'ordre décroissant. PATCH affiche ensuite les numéros des secteurs sur lesquels il effectue la recherche jusqu'à ce que celle-ci soit satisfaite. Une fois la chaîne trouvée, le secteur sur laquelle elle se trouve est affiché (zones 1, 2, 3, mises à jour), la ZONE 4 est effacée, et le curseur se positionne sur le premier caractère de la chaîne.

Si la chaîne n'a pas été trouvée, PATCH affiche le message "Chaîne non Trouvée" attend un <rc> de la part de l'utilisateur puis affiche le dernier secteur qui a été consulté (zones 1, 2, 3 mises à jour). La ZONE 4 est effacée et le curseur est positionné dans le coin en haut à gauche de la ZONE 2.

Remarques :

- La recherche de chaîne sera effective, si le curseur ne pointe pas déjà sur le premier caractère de la chaîne.
- Si la chaîne trouvée est à cheval sur deux secteurs, c'est le secteur où se trouve le premier caractère de la chaîne qui est visualisé.
- Il est possible d'interrompre la recherche d'une chaîne en frappant la touche "ESC". Dans ce cas, PATCH visualise le dernier secteur lu. Les zones 1, 2, 3 sont mise à jour en conséquence, le curseur se place dans le coin en haut à gauche de la ZONE 2 et la ZONE 4 est effacée.

* **N : Nouvelle Recherche**

Cette commande permet de répéter la dernière commande "K" exécutée. Seule la confirmation du sens de la recherche est demandée. Le reste est identique à la demande "K".

* **I : Imprimer le Secteur en cours**

Cette commande permet d'afficher sur l'imprimante le secteur visualisé tel qu'il apparaît sur l'écran. Si des modifications ont été faites, elles seront imprimées. Ce n'est donc pas le secteur tel qu'il est sur le support qui est imprimé.

Pendant l'impression, le curseur viendra se placer dans la ZONE 4. Une fois l'impression terminée, il reprendra sa position initiale.

Remarque :

L'impression du secteur ne peut se faire que si votre système possède le type d'impression "hard-copy".

* **S : Impression enchaînée**

Cette commande permet d'imprimer plusieurs secteurs à partir du secteur visualisé et en incrémentant le numéro du secteur. Chaque secteur est visualisé puis imprimé immédiatement comme avec la commande "I". L'arrêt de l'impression des secteurs se fera sur fin de volume, si nous avons utilisé la syntaxe (1) et sur fin de fichier si nous avons utilisé la syntaxe (2). Mais, il est possible d'arrêter l'impression enchaînée par la frappe d'un <rc>. L'impression sera alors interrompue après que le secteur visualisé ait été imprimé. Les zones 1, 2, 3 seront mises à jour et le curseur se placera dans le coin en haut à gauche de la ZONE 2.

* **R : Visualisation enchaînée**

Cette commande permet de visualiser plusieurs secteurs, les uns après les autres, à partir de celui qui est visualisé et en incrémentant le numéro du secteur. L'arrêt de l'affichage des secteurs se fera sur fin de volume, si nous avons utilisé la syntaxe (1) et se fera sur fin de fichier si nous avons utilisé la syntaxe (2). Mais il est possible d'arrêter la visualisation enchaînée en frappant n'importe quelle touche. PATCH restera alors sur le dernier secteur visualisé. Les zones 1, 2, 3 seront mises à jour et le curseur se placera dans le coin à gauche dans la ZONE 2.

* **Z : Remplissage du secteur avec une valeur**

Cette commande permet d'initialiser un secteur avec une valeur hexadécimale quelconque. Après avoir frappé la commande, PATCH affiche le message "Valeur hexadécimale :" puis attend que l'utilisateur frappe une valeur suivie de <rc>. Ensuite, PATCH remplira la ZONE 2 du caractère tel qu'il a été entré et remplira la ZONE 3 de son équivalent ASCII. En aucun cas le support n'est affecté. Si vous souhaitez réécrire le secteur sur le support vous devez utiliser la commande "M".

Une fois la commande "Z" exécutée, la ZONE 4 sera effacée et le curseur se positionnera dans le coin en haut à gauche de la ZONE 2.

Si l'utilisateur frappe uniquement <rc> ou frappe un caractère qui n'est pas hexadécimal, la commande "Z" est annulée. La ZONE 4 est alors effacée et le curseur reprend sa place initiale dans la ZONE 2.

* **X : Convertir une chaîne en hexadécimal**

Cette commande permet d'avoir l'équivalent en hexadécimal d'une chaîne de caractères ASCII. Elle peut être utilisée lorsqu'il faut modifier le contenu d'un secteur et qu'on ne connaît pas la valeur hexadécimale des codes à insérer. Après la frappe de la commande, PATCH envoie le message "Chaîne à convertir :" dans la ZONE 4 et attend une chaîne suivie d'un <rc>. Au fur et à mesure que l'utilisateur entre un caractère, apparaît en-dessous de celui-ci son codage hexadécimal. La frappe d'un <rc> provoque un retour du curseur à sa place initiale dans la ZONE 2. La ZONE 4 n'est pas effacée afin que l'utilisateur puisse éventuellement modifier le secteur visualisé avec les codes hexadécimaux des caractères décodés dans la ZONE 4.

Remarque :

La commande "X" n'affecte pas le secteur en cours ni sa visualisation.

* **Esc : Sortir de PATCH**

La frappe de la touche ESC provoque le retour à l'interpréteur de commandes. En aucun cas cette commande n'affecte le support et donc le dernier secteur visualisé n'est pas réécrit.

La commande ESC ne peut être activée que lorsque le curseur se trouve dans la ZONE 2.

4.3.2. **Etat d'un support périphérique : "STATUS"**

Présentation

En cours d'exploitation, tout incident sur les périphériques sectorisés (disques, disquettes...) est enregistré par le système dans des tables réservées à cet usage. Ces tables sont initialisées à la mise sous tension et à chaque chargement du système.

Avec STATUS vous pouvez à chaque instant consulter ces tables et ainsi vérifier l'état de vos unités sectorisées. C'est donc un outil d'aide au diagnostic qui vous permet de surveiller la détérioration de vos volumes.

L'utilitaire STATUS peut être installé, de manière permanente, en mémoire lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant, sauf si le programme est déjà utilisé par un autre poste (STATUS est installable en mémoire). Cette installation se fait par la déclaration suivante dans le fichier source "SYSCONF-S" :

— STATUS < INTR

Reportez-vous au chapitre 7 (Configuration des utilitaires résidents), pour plus de détails.

Commande d'appel

—> [unité1.]STATUS[,unité2][,LIS=LO]<rc>

unité1 indique l'unité sur laquelle STATUS est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire STATUS est recherché sur l'unité implicite programme de votre poste.

STATUS nom du fichier binaire correspondant ayant comme suffixe "X".

unité2 indique l'unité à diagnostiquer. Par défaut, l'unité sélectionnée sera implicite fichier de votre poste.

Exemple :

—> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0

—> STATUS

Status des périphériques sectorisés Version 1.8b

Copyright (c) 1983 BULL Micral

SUPPORT MD.0

ECHANGES	ERR LEC.	EERR ECR.	ERR POSIT.	SECT. IV.
69595	0	0	0	0

—> _

Nous avons ici l'état de l'unité implicite fichier, c'est-à-dire l'unité "MD0". Il a été fait 69595 accès à cette unité et aucune erreur n'a été détectée par le système depuis le dernier chargement de celui-ci.

Messages d'exécution

STATUS affiche plusieurs messages dont les informations ont la signification suivante :

SUPPORT nom de l'unité examinée

ERR xx numéro de la dernière erreur rencontrée (xx est le numéro d'erreur). Ce message ne sera pas affiché s'il n'y a pas eu d'erreur. Il est suivi du message "EN ECT." si l'erreur a eu lieu lors d'une écriture et de "EN LEC." si l'erreur a eu lieu lors d'une lecture.

NO PISTE numéro de piste sur laquelle a été provoquée la dernière erreur. Ce message ne sera pas affiché s'il n'y a pas eu d'erreur.

NO SECTEUR numéro de secteur, de la piste ci-dessus nommée, sur lequel a été provoquée l'erreur. Ce message ne sera pas affiché s'il n'y a pas eu d'erreur.

ECHANGES nombre total d'opérations de lecture et d'écriture depuis le dernier chargement du système.

ERR LEC. nombre total d'erreurs rencontrées en lecture depuis le dernier chargement.

- ERR ECR. nombre total d'erreurs rencontrées en écriture depuis le dernier chargement.
- ERR POSIT. nombre total d'erreurs de positionnement de piste. Cette erreur est à différencier des erreurs de lecture et d'écriture. Elle indique que la piste sur laquelle devait s'effectuer la lecture ou l'écriture, n'a pu être atteinte.
- SECT. INV. uniquement avec des disques durs. Nombre total de secteurs invalides donc inutilisables. C'est une information permanente qui n'est pas réinitialisée lors du chargement du système.
- LISTE PISTES/SECTEURS INVALIDES : uniquement avec des disques durs. C'est la liste des secteurs qui ont été substitués. Ils sont énumérés sous la forme "P/S" où P représente le numéro de piste et S le numéro de secteur dans cette piste.

Remarques :

- Une erreur de lecture ou d'écriture peut avoir deux causes : fonctionnement défectueux de l'unité de lecture/écriture ou support disque détérioré. Il est de votre responsabilité de tirer parti des renseignements fournis par la commande STATUS pour établir votre diagnostic et prendre les dispositions propres à améliorer les conditions de votre exploitation.
- Toute anomalie de lecture ou d'écriture donne systématiquement lieu à des tentatives d'accès répétées qui ne se traduisent pas nécessairement par des erreurs. Ces tentatives qui s'opèrent à votre insu, sont néanmoins enregistrées et signalées par STATUS. Il en découle que les statistiques obtenues peuvent faire état d'un nombre d'incidents supérieur à celui que vous aurez observé.
- Sur les disques durs, des secteurs de remplacement se substituent aux secteurs invalides. Ces substitutions s'effectuent ici encore à votre insu. L'utilisation de la commande STATUS vous permettra de vérifier néanmoins que le nombre de secteurs invalides n'augmente pas dans des proportions suspectes.

Exemple 1 :

—> STATUS,FL0

Status des périphériques sectorisés Version 1.8b

Copyright (c) 1983 BULL Micral

SUPPORT FL.0 ERR 02 EN ECT. NO PISTE 91 / NO SECT 10

ECHANGES	ERR LEC.	ERR ECR.	ERR POSIT.	SECT. INV.
3550	5	26	6	0

—> _

STATUS nous montre l'état de l'unité "FL0". Depuis le dernier chargement du système, il y a eu 3550 accès de lecture ou d'écriture à cette unité. 5 erreurs de lecture, 26 erreurs d'écriture et 6 erreurs de positionnement ont été détectées. La dernière erreur, est une erreur 02 en lecture sur le secteur 0 de la piste 91.

Exemple 2 :

—> STATUS, MD0

Status des périphériques sectorisés Version 1.8b

Copyright (c) BULL Micral

SUPPORT MD.0

ECHANGES	ERR LEC.	ERR ECR.	ERR POSIT.	SECT. INV.
69595	0	0	0	0

** LISTE PISTES/SECTEURS INVALIDES :

1314/29 1332/20

—> _

STATUS nous montre l'état de l'unité "MD0". Depuis le dernier chargement du système, il n'y a eu aucune erreur détectée, ni en lecture, ni en écriture, ni en positionnement. Il nous signale que sur le VOLUME (et non sur le support), il y a deux secteurs invalides. Ceux-ci sont énumérés en-dessous :

il y a le lecteur 29 de la piste 1314 et le secteur 20 de la piste 1332. Ces deux secteurs ont été substitués par d'autres par le système.

Exemple 3 :

-> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0

-> STATUS,FL0

Status des périphériques sectorisés Version 1.8b

Copyright (c) 1983 BULL Micral

SUPPORT FL.0

ECHANGES	ERR LEC.	ERR ECR.	ERR POSIT.	SECT. INV.
0	0	0	0	0

-> /,FL0

Catalogue du volume Version 3.4b

Copyright (c) 1983 BULL Micral

Erreur Périphérique n° 2 Erreur Lecture/Ecriture

-> STATUS,FL0

Status des périphériques sectorisés Version 1.8b

Copyright (c) 1983 BULL Micral

SUPPORT FL.0 ERR 02 EN LEC. NO PISTE 0 / NO SECT 4

ECHANGES	ERR LEC.	ERR ECR.	ERR POSIT.	SECT. INV.
0	5	0	0	2

-> _

La première commande STATUS nous montre que l'unité de disques souples "FLO" a été inutilisée depuis le dernier chargement du système (nombre d'échanges à zéro). La commande qui suit provoque une erreur 2. A priori, nous ne pouvons pas savoir si cette erreur provient du disque dur "MD0", au moment du chargement du fichier "/-X", ou du disque souple "FLO" au cours de l'exécution de l'utilitaire "/". La deuxième commande STATUS nous renseigne immédiatement, en nous signalant une erreur 2 sur le secteur 4 de la piste 0. On peut remarquer, sur le résultat de ce deuxième STATUS, que le nombre d'échanges est à zéro. Cela signifie que l'erreur s'est produite sur le premier accès. Il est aussi signalé 5 erreurs de lecture. Cela n'est pas en contradiction avec l'abandon au premier échange, mais cela veut dire que le système a fait 5 tentatives d'accès, sans succès, à l'unité "FLO". Si la dernière tentative n'avait pas échoué, l'erreur 2 ne serait pas apparue. Un STATUS après ce dernier échange nous aurait affiché : un nombre d'échanges égal à 1 et un nombre d'erreurs de lecture égal à 4.

Messages d'erreurs

Hormis un message

"**imprimante non prête, sélectionnée (o/n) ? :**",
affiché lorsqu'on désire avoir les résultats sur l'imprimante et que celle-ci n'est pas disponible, STATUS n'envoie pas de messages particuliers. Si l'unité sur laquelle on désire avoir les informations n'est pas un support sectorisé ou est inexistante, STATUS retourne une erreur système 30.

Exemple 1 :

```
-> STATUS,MD0,LIS=LO  
imprimante non prête, sélectionnée (o/n) ? : N <rc>  
    Erreur Périphérique n° 1 Unité Non Prête  
-> _
```

Après avoir lancé un STATUS sur le disque dur "MD0", l'utilitaire lui a signalé que l'imprimante n'était pas prête. L'utilisateur a abandonné et l'erreur système 1, relative à l'état de l'imprimante, a été affichée.

Exemple 2 :

- > Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0
- > STATUS,IM0
Erreur de Commande n° 30 Syntaxe Incorrecte
- > STATUS,FL1
Erreur de Commande n° 30 Syntaxe Incorrecte
- > _

Dans le premier cas, la commande est refusée car "IM0" n'indique pas un support sectorisé.

Dans le deuxième cas, le système n'ayant qu'une unité de disques souples l'unité "FL1" ne peut être atteinte.

4.3.3. Visualisation des partitions mémoire : "STMEM"

Présentation

En cours d'exploitation, notamment dans une configuration multiposte, il peut être intéressant de visualiser l'état de la mémoire tout d'abord au repos, c'est-à-dire lorsqu'aucun poste ne travaille (donc sous l'interpréteur de commandes) et qu'aucune tâche de fond n'est active (par exemple le "SPOULEUR"), puis de visualiser l'état de la mémoire lorsque les différents postes sont actifs. On peut ainsi comparer les deux états et voir le taux d'occupation de la mémoire, quels sont les postes ou tâches qui prennent le plus de place.

Commande d'appel

—> [unité1.]STMEM<rc>

où :

unité1 indique l'unité sur laquelle l'utilitaire STMEM est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire STMEM est recherché sur l'unité implicite programme de votre poste.

STMEM nom du fichier binaire correspondant ayant comme suffixe "X".

Remarque :

La commande STMEM n'admet aucun paramètre. C'est donc toute la mémoire qui est analysée.

Les différentes options de STMEM

Une fois la commande lancée et acceptée par l'interpréteur de commandes, STMEM affiche sa bannière et un menu de la manière suivante :

—> STMEM

Etat de la mémoire Version 1.2a

Copyright (c) 1983 BULL Micral

- * Sortie (ESC)
- * Aide (0)
- * Etat de la mémoire en mode continuuel (1)
- * Etat de la mémoire (Autre touche)
puis retour au système

Votre choix : —

L'utilisateur se trouve alors devant un choix entre quatre réponses possibles :

- En frappant sur la touche ESC (ou SORTIE), il revient à l'interpréteur de commandes.
- En frappant le caractère numérique "0", STMEM affiche une documentation sommaire commentant ses propres messages (STMEM est auto-documenté).
- En frappant le caractère numérique "1", STMEM affiche l'état de la mémoire d'une manière permanente, c'est-à-dire qu'à chaque modification de cet état, un réaffichage aura lieu.
- En frappant toute autre touche, STMEM affiche l'état de la mémoire puis revient à l'interpréteur de commandes.

L'état de la mémoire

En frappant sur les touches numériques "0" ou "1", on fait apparaître sur le terminal un tableau dont chaque ligne représente un segment mémoire et son état. A tout moment, cet affichage peut être interrompu en frappant la touche ESC (ou SORTIE). On peut alors revenir à l'interpréteur de commandes en frappant la touche <rc> ou continuer l'affichage en frappant de nouveau sur la touche ESC (ou SORTIE).

Voici l'exemple d'un tableau affiché :

POSTE	BASE	LONGUEUR	PARTITION	TYPE
0	0B42	0010	S	E
0	0B52	0004	S	E
1	0B56	0010	S	E
0	0B66	0004	S	E
	0B6A	00D8	S	
0	0C42	003B	0	I
	0DFD	03C5	0	
1	1042	0180	1	E
	11C2	0280	1	
	1442	0012	x	
0	1454	0520	x	E
0	1974	080A	x	I
	217E	579D	x	
0	791B	00A0	x	E
0	79BB	0180	x	E
0	7B6B	0040	x	E
0	7BAB	0200	x	E
0	7DAB	0035	x	E
1	7DED	0010	x	E
0	7DF0	0010	x	E

*** Colonne "POSTE" :**

Elle indique le numéro du poste qui a alloué le segment et qui en est donc le propriétaire. Seul ce poste peut le désallouer. Si aucun numéro de poste n'est spécifié, cela signifie que le segment est libre et donc disponible pour tout poste ou toute tâche de fond.

*** Colonne "BASE" :**

Elle indique l'adresse de début du segment. Cette adresse, exprimée en hexadécimal est un numéro de paragraphe (rappelons qu'un paragraphe comprend 16 octets).

*** Colonne "LONGUEUR" :**

Elle indique la longueur alloué ou libre et est exprimée en hexadécimal (nombre de paragraphes).

* **Colonne "PARTITION" :**

Elle indique à quelle partition appartient le segment :

S = partition système
0 à n = partition réservée au poste "n"
x = partition commune

* **Colonne "TYPE" :**

Elle indique le type d'allocation pour le segment concerné. Si rien n'est mentionné, le segment est libre et donc disponible.

I = allocation implicite. Le segment est automatiquement désalloué quand on revient à l'interpréteur de commandes.

E = allocation explicite. Le segment n'est libéré que sur commande explicite du programme.

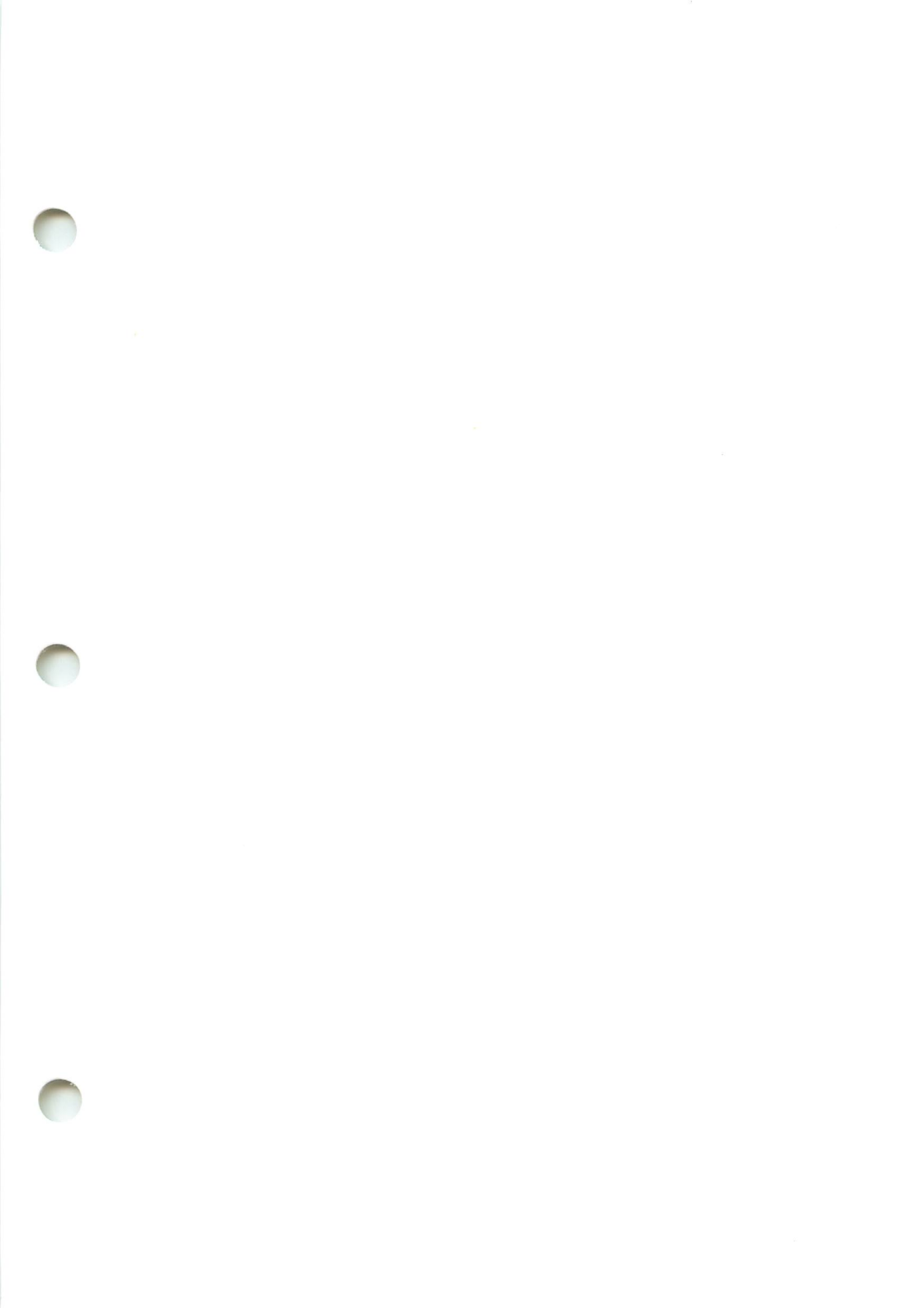
Messages d'erreur

STMEM n'affiche pas de message d'erreur. Il ne vérifie pas la cohérence de la table des segments mémoire.

Remarques et conseils

Lorsqu'au cours de l'exploitation, une erreur système 27 ou 38 se produit, il est intéressant d'utiliser STMEM pour voir l'espace mémoire disponible et l'espace mémoire occupé par les différents programmes. Ainsi, en fonction de l'espace demandé par votre programme qui n'a pu être exécuté, vous serez amené à supprimer certains modules d'extension inutilisés du configurateur ou supprimer des utilitaires résidents qui ne servent que trop rarement et qui se trouvent sur votre unité implicite programme.

STMEM vous permet de vous assurer que, lorsque tous les postes sont rendus inactifs ainsi que les tâches de fond, tous les espaces mémoires occupés par les différents programmes ont bien été libérés. Mais attention, il faut être prudent dans l'interprétation du résultat, car certains programmes lors de leur premier chargement en mémoire initialisent des tables qui deviennent résidentes. C'est le cas des décors, notamment.



5.	L'EDITEUR DE TEXTES : "EDV"	5-1
5.1.	Introduction	5-1
5.1.1.	Présentation générale	5-1
5.1.2.	Modes de travail	5-2
5.1.3.	Touches spécialisées	5-2
5.1.4.	Présentation de fichiers	5-3
5.2.	Chargement de EDV	5-4
5.3.	La gestion des fichiers de l'éditeur de texte	5-5
5.3.1.	Le fichier origine	5-5
5.3.2.	Les fichiers intermédiaires	5-5
5.3.3.	Le fichier destination	5-7
5.4.	Gestion de l'écran	5-8
5.4.1.	Les trois zones de l'écran	5-8
5.4.2.	Occupation de EDV en mémoire	5-8
5.5.	Les modes de fonctionnement	5-9
5.5.1.	Le mode vidéo	5-9
5.5.2.	Le mode commande	5-9
5.5.3.	Entrée/sortie des modes de l'éditeur	5-9
5.6.	Présentation et utilisation du mode vidéo	5-10
5.6.1.	Présentation en création de fichier	5-10
5.6.2.	Présentation en mise à jour de fichier	5-11
5.6.3.	Utilisation en création ou mise à jour de fichier	5-12

5.7.	Les procédés de saisie de caractères	5-13
5.8.	Les commandes en mode vidéo	5-14
5.8.1.	Les mouvements fichier	5-14
5.8.2.	Les mouvements curseur	5-15
5.8.2.1.	Curseur début/fin écran et début/fin ligne	5-15
5.8.2.2.	Les déplacements ligne	5-15
5.8.2.3.	Les déplacements par mot	5-16
5.8.2.4.	Les déplacements par caractère	5-17
5.8.3.	Les suppressions	5-18
5.8.3.1.	Au niveau de la ligne	5-18
5.8.3.2.	Au niveau du mot	5-18
5.8.3.3.	Au niveau du caractère	5-19
5.8.3.4.	Commande d'insertion	5-19
5.8.3.5.	Les autres commandes	5-19
5.8.4.	Présentation et utilisation en mode de fonctionnement commande	5-21
5.8.4.1.	Commandes de sauvegarde du fichier	5-21
5.8.4.2.	Commandes diverses	5-23
5.8.4.3.	Les commandes de déplacement du pointeur et du curseur	5-24
5.8.4.4.	Les commandes de suppression de texte	5-25
5.8.4.5.	Les commandes d'insertion	5-26
5.8.4.6.	Commandes de recherche de chaînes de caractères	5-26
5.8.4.7.	Zone de travail pour manipuler des parties du fichier	5-28
5.8.4.8.	Mouvements de fichiers	5-31
5.8.4.9.	Les boucles	5-31

5. L'EDITEUR DE TEXTES : "EDV"

5.1. INTRODUCTION

EDV est l'éditeur de texte de BULL MICRAL pour ordinateurs 16 bits qui supportent le système PROLOGUE. Il vous permet de créer et mettre à jour du texte.

5.1.1. Présentation générale

Cet éditeur possède un affichage rapide et agréable lors du défilement ligne par ligne.

L'utilisateur travaille sur tout l'écran : c'est un éditeur pleine page.

Le curseur, symbolisé par un tiret, vous permet de visualiser vos déplacements.

La répétition d'ordres ou de commandes de l'éditeur est facilitée par la mise en mémoire de la dernière commande validée :

il suffit de faire un retour chariot pour revalider l'ordre ou la commande autant de fois qu'il est nécessaire.

Ce produit possède également une notion originale de ligne :

- La ligne logique : validée par un retour chariot.
Sa taille est limitée par celle des fichiers utilisés par EDV.
- la ligne écran : limitée par les caractéristiques de l'écran.

L'ensemble des lignes logiques constitue le fichier traité par l'éditeur.

5.1.2. Modes de travail

Pour travailler vous disposez de deux modes :

Mode video : il permet la saisie et la mise à jour de votre fichier en visualisant une partie de celui-ci à l'écran.
Ceci permet de voir les déplacements du curseur et les modifications effectuées.

Le texte est traité page par page.

Le fichier texte peut être parcouru dans deux sens :

- page suivante, ligne suivante
- page précédente, ligne précédente

Le curseur se déplace verticalement et horizontalement dans les deux sens.

Les suppressions se font à différentes échelles :

- ligne logique entière
- début ou fin de ligne
- mot
- caractère

L'insertion se fait caractère par caractère.

mode commande : il permet la sauvegarde, la fin de mise à jour, le positionnement en début de fichier ou fin de fichier, la recherche de chaînes de caractères, les déplacements de chaînes, les répétitions de commandes, l'insertion de fichier...

5.1.3. Touches spécialisées

Il existe également des touches spécialisées :

— la touche contrôle (CTRL) : elle différencie les lettres introduites dans le texte des lettres générant une commande. Cette touche est représentée par “^”.

Généralement cet accent est suivi d'une lettre de commande. Par exemple : ^A

— la touche retour chariot (RC) : elle permet de passer à la ligne physique suivante ou de terminer une ligne texte (ligne logique).

- la touche escape (ESC) : elle sert à passer en mode commande ou à interrompre une commande telle une recherche de chaîne de caractères dans un fichier. Sa représentation est "ESC". Sur l'écran elle est visualisée par "[".
- la touche effacement : elle permet de supprimer un caractère à gauche du curseur.
- la touche tabulation : elle permet un déplacement à droite de huit caractères.

5.1.4. Présentation de fichiers :

L'éditeur de texte a deux types de fichiers apparents.

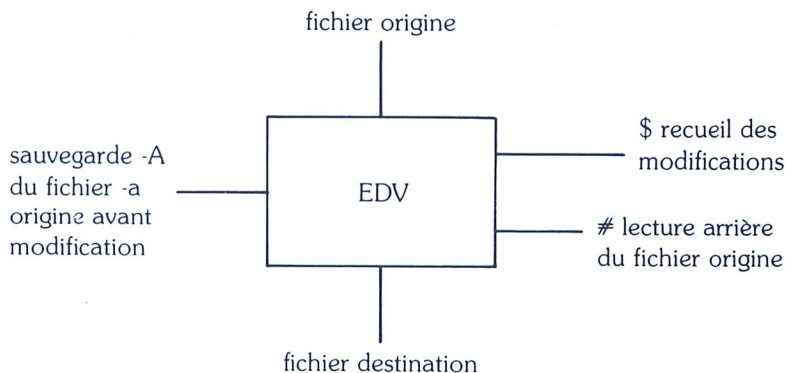
Un premier fichier que l'on nomme lors de la création et qui possède le type choisi par l'utilisateur : un à trois caractères acceptés par le système PROLOGUE.

C'est le fichier origine.

Ce type doit être différent de \$, #, a, A qui sont réservés à EDV.

Un fichier destination, non obligatoire, dont le type est attribué par l'utilisateur. Au cas où il y aurait une erreur de fonctionnement, EDV restitue le fichier origine non modifié en cas de mise à jour.

En création, tous les fichiers sont abandonnés et devront être recréés.



5.2. **Chargement de EDV :**

Objet de la commande :

Pour exploiter EDV, il est nécessaire de le charger en mémoire.

Syntaxe :

—>[unité.]EDV, nomfich1[,DEST=nomfich2] <rc>

où :

unité : c'est le nom de l'unité disque sur laquelle se trouve EDV. Si il n'est pas précisé, EDV prend l'implicite système.

nomfich : c'est le fichier origine sur lequel on veut travailler.

Vous pouvez écrire en majuscule ou en minuscule tous les caractères acceptés par le système PROLOGUE.

nomfich2 : fichier de destination où l'on sauvegarde la mise à jour du travail faite à partir de nomfich1.

L'ensemble DEST = nomfich2 est optionnel. Il permet de créer à partir d'un fichier source existant, un autre fichier source sans détruire le fichier origine.

nomfich2 doit toujours être précédé de : DEST=

En création de fichier origine, nomfich2 est ignoré par EDV.

L'éditeur affiche sous la bannière : nouveau fichier

Remarque :

Si dans l'option "DEST=nomfich2" seul le nom de l'unité figure, l'éditeur prend par défaut DEST=unité2.nomfich1 et compare les caractéristiques des fichiers nomfich1 et unité2.nomfich1.

— si nomfich1 et unité 2.nomfich1 sont identiques EDV revient au système PROLOGUE. Le fichier origine n'est pas modifié.

— si la disquette du fichier destination est protégée en écriture. EDV affiche sa bannière.

Ensuite il revient au système PROLOGUE.

5.3. La gestion de fichiers de l'éditeur de texte

L'éditeur de texte travaille avec trois fichiers :

- le fichier origine
- le fichier destination
- les fichiers intermédiaires

5.3.1. Le fichier origine

Le nom du fichier est celui déclaré à la création par "nomfich1". Sur le fichier origine, EDV ne fait que des accès lecture sur le fichier origine. En fin d'exécution le type de ce fichier est changé si l'utilisateur n'a pas précisé l'option DEST.

Le nouveau type sera A (voir schéma).

Si en cours d'exécution une erreur quelconque d'accès sur l'unité se produit, le fichier origine est conservé avec ses caractéristiques (support, nom, type et clés d'accès).

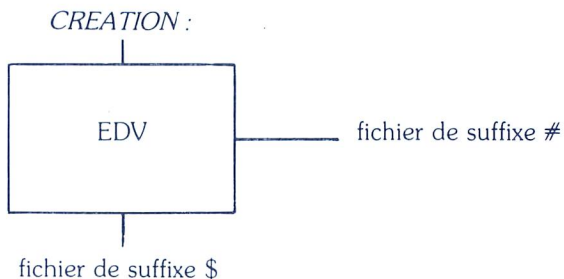
Si vous n'avez pas précisé le type du fichier origine., EDV, recherche un fichier de même nom et de type "S". S'il le trouve, il travaille avec celui-ci sinon il crée un fichier avec le support, le nom, et les clés donnés par l'utilisateur.

5.3.2. Les fichiers intermédiaires

Les caractéristiques des fichiers intermédiaires sont les suivantes :

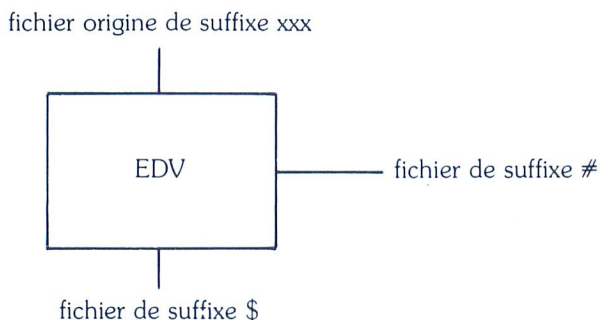
- unité identique à celle du fichier destination
- nom identique à celui du fichier destination
- suffixe "#" ou "\$"
- clés d'accès identiques à celles du fichier destination

Schéma récapitulatif de la gestion des fichiers :



Quand EDV est appelé, deux fichiers de suffixe # et \$ sont aussitôt créés. En fin de création, le fichier de suffixe \$ prend le type —xxx choisi par l'utilisateur lors de l'appel de EDV. Le fichier de type # est détruit.

MODIFICATION SANS FICHIER DESTINATION :



En fin de travail, le fichier de suffixe # est supprimé. Le fichier de suffixe xxx, contient donc les dernières modifications.

Ces fichiers sont exclusivement gérés par l'éditeur.

En fin d'exécution ces fichiers sont refermés et supprimés. Le fichier suffixe # ne sert que pour les mouvements arrière à l'intérieur du fichier et le suffixe \$ pour les modifications.

En fonctionnement régulier et normal, l'utilisateur n'a pas connaissance de ces fichiers. Ils n'apparaissent sur aucun catalogue. De plus ces fichiers ne sont pas récupérables par l'utilisateur.

5.3.3. Le fichier destination

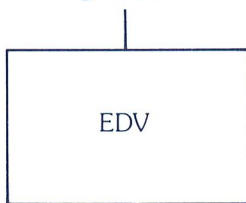
Les caractéristiques du fichier destination sont, soit données dans la commande par l'utilisateur, soit données par l'éditeur si l'option nomfich2 n'est pas précisée dans la commande.

Les caractéristiques du fichier destination données par EDV sont :

- unité identique à celle du fichier origine
- nom identique à celui du fichier origine
- suffixe "\$" en cours d'exécution de l'éditeur
- clés d'accès identiques à celles du fichier origine

L'éditeur renomme le fichier origine en suffixe A.

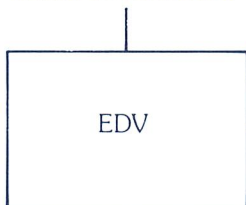
fichier origine de suffixe -A



fichier origine de suffixe -xxx

MODIFICATION AVEC OPTION DESTINATION :

fichier de suffixe xxx



fichier de suffixe #

fichier de suffixe yyy

Après l'édition, le fichier de suffixe # est supprimé. Les deux autres sont fermés.

Vous avez ainsi sur votre support magnétique le fichier origine de suffixe xxx et votre fichier destination de suffixe yyy qui contient les dernières modifications.

5.4. GESTION DE L'ECRAN

EDV travaille sur le nombre de lignes et de colonnes déclarées dans le système. Il s'adapte aux limites physiques imposées par ce dernier. Il en est de même pour l'affichage des caractères.

5.4.1. Les trois zones de l'écran

- la ligne d'état où se trouve écrit, le nom du fichier, le mot "INSERTION" si on est en train de l'utiliser, et une étoile si on est en mode commande. Cette zone prend deux lignes à écran.
- la zone "aide utilisateur" qui peut être affichée à la demande de l'utilisateur.
(^ J:cf.mode vidéo et ^ Q:cf.mode commande). Elle donne à ce dernier un rappel des différentes commandes en mode vidéo. Cette zone occupe huit lignes à l'écran.
- la zone d'affichage de la page du fichier.
Elle se compose suivant la présence de la zone précédente du nombre de lignes disponibles à l'écran moins deux ou dix lignes.

5.4.2. Occupation de EDV en mémoire

EDV prend environ 21 Koctets en mémoire, plus la place des fichiers traités par EDV qui varie de 4 à 32 K. Il faut rajouter 1 K pour les données propres à EDV.

5.5. LES MODES DE FONCTIONNEMENT

L'éditeur EDV possède deux modes de fonctionnement :

- le mode vidéo
- le mode commande

5.5.1. Le mode vidéo

Après avoir chargé EDV, l'utilisateur se trouve toujours en mode vidéo. Ce mode de fonctionnement permet à l'utilisateur de créer ou de mettre à jour son fichier source en le visualisant. L'opérateur peut suivre à l'écran tous les déplacements du curseur à l'intérieur du fichier. Il peut ainsi voir toutes les créations et modifications qu'il est en train d'exécuter.

5.5.2. Le mode commande

Le mode commande permet des commandes plus puissantes : sauvegarde fichier, positionnement début/fin de fichier, recherche de chaînes de caractères, ...

5.5.3. Entrée/sortie des modes de l'éditeur

L'éditeur démarre toujours en mode vidéo.

Pour passer en mode commande à partir du mode vidéo, il suffit d'appuyer sur la touche ESC.

Pour passer du mode commande en mode vidéo il faut frapper "V<rc>" ou terminer sa commande par "V<rc>".

5.6. PRESENTATION ET UTILISATION DU MODE VIDEO

5.6.1. Présentation en création de fichier

L'utilisateur frappe au clavier de la commande :

—>edv, nomfich1<rc>

L'option dest=nomfich2 est inutile dans le cas d'une création de fichier. Il affiche la bannière avec l'intitulé "nouveau fichier" puis l'en-tête avec une page vide.

Le curseur se positionne à la première colonne de la première ligne de cette page.

L'utilisateur peut à partir de ce moment créer son fichier.

premier écran

—>edv, nomfich1<rc>
Editeur de texte Version X.X
***** #####
Copyright (c) 1984 Bull Micral
nouveau fichier —

deuxième écran

— fichier : XXXXXXXX-YYY
—

Le tiret matérialise le curseur.
La chaîne de caractères XXXXXXXX-YYY représente le nom et le type du fichier en cours de création.

5.6.2. Présentation en mise à jour de fichier

L'utilisateur frappe au clavier la commande :

->edv,nomfich1[,dest=nomfich2]<rc>
premier écran

<pre>->edv,nomfich1[,dest=nomfich2]<rc> Editeur de texte Version X.X ***** ##### Copyright (c) 1984 Bull Micral</pre>
--

deuxième écran

fichier : XXXXXXXX-YYY

- titre "programme exemple version 1.0 du 12/12/83
--

Le tiret représente le curseur.

Le nom XXXXXXXX-YYY représente le nom et le type du fichier.

L'utilisateur peut à partir de ce moment faire la mise à jour du fichier.

5.6.3. Utilisation en création ou mise à jour de fichier :

L'utilisateur peut à partir de ce moment appuyer sur toutes les touches désirées. Elles auront un fonctionnement bien particulier pour chacune d'elles.

Il y a deux sortes de caractères en mode vidéo :

- les caractères faisant partie du texte
- les caractères générant une commande

La touche CTRL est la "clé" de l'éditeur. chaque accent circonflexe précédant un caractère, implique l'enfoncement simultané de la touche CTRL suivi de celle du caractère. Ainsi pour saisir la commande " ^W", il faut appuyer sur la touche CTRL, et sans la relacher appuyer sur "W".

Attention à la commande "^^" qui signifie contrôle accent circonflexe : il faut appuyer sur la touche contrôle et sur l'accent circonflexe.

La touche contrôle différencie les caractères générant une commande de ceux faisant partie du texte, à part trois exceptions : la touche effacement, la touche sortie (ESC), et le retour chariot <rc>.

5.7.

LES PROCEDES DE SAISIE DE CARACTERES

L'utilisateur frappe au clavier des caractères faisant partie du texte :

- soit, ils sont insérés dans le texte
- soit, ils remplacent d'autres caractères

Au niveau opérateur :

En appuyant sur ^O le procédé de saisie des caractères change.

En procédé modification il n'y a aucune indication dans la ligne d'état. Par contre si l'opérateur frappe ^O le mot insertion s'affiche sur cette ligne. Dans ce cas tous les caractères faisant partie du texte sont insérés dans ce dernier sans suppression de caractères présents.

Si l'utilisateur frappe à nouveau sur ^O il revient en procédé modification.

Au niveau de la ligne texte :

Si le caractère frappé intervient en milieu de ligne, il vient en remplacement de celui qui se trouve à la place du curseur.

Par contre, s'il intervient en fin de ligne de texte il est inséré après le dernier caractère de ce texte.

Dans ce procédé de saisie la frappe de <rc> a pour effet de positionner le curseur en début de ligne texte suivante.

5.8. LES COMMANDES EN MODE VIDEO

5.8.1. Les mouvements fichier

Ces commandes ne modifient pas le contenu du fichier. Elles ont pour but de visualiser le fichier à l'écran.

^C

Visualisation de la page écran suivante.

Le curseur est positionné en début de page. En fin de fichier si l'opérateur continue à frapper cette commande, le curseur se positionne au début de la dernière ligne et cette commande est rendue inactive tant que le curseur est positionné sous la dernière ligne du fichier.

^R

Visualisation de la page écran précédente.

Le curseur est positionné au début de la dernière ligne de la page.

En début de fichier si l'opérateur demande l'exécution de cette commande le curseur se positionne au début de la première ligne de la page et cette commande est rendue inactive tant que le curseur est positionné sur la première ligne du fichier.

^Z

Le fichier remonte d'une ligne écran.

Le curseur reste positionné sous le même caractère. Si celui-ci disparaît de l'écran le curseur se positionne à l'intérieur de la première ligne de la page. En fin de fichier la dernière ligne reste affichée et la commande est rendue inactive tant qu'il n'y a pas de nouvelle ligne.

^W

Le fichier descend d'une ligne écran

Le curseur reste positionné sous le même caractère. Si ce caractère disparaît, il réagit comme ci-dessus. En début de fichier la commande est inactive tant que la première ligne du fichier est en tête d'écran.

^L Ligne repérée par le curseur, positionnée en milieu de page écran.

La ligne où se trouve le caractère pointé par le curseur, se place en milieu d'écran.

Si le caractère est dans la première moitié de la première page du fichier, il n'y a aucun changement au niveau de l'affichage.

5.8.2. Les mouvements curseur

5.8.2.1. Curseur début/fin écran et début/fin ligne

^ ^ Curseur en début ou en fin d'écran

Le curseur se place toujours en début de ligne.

La commande est frappée une fois, le curseur se place en début d'écran.

Une deuxième fois, le curseur se place sur le premier caractère de la dernière ligne écran.

^B Curseur en début ou en fin de ligne physique écran.

La commande est frappée une première fois le curseur se place en début de ligne.

Une deuxième fois, le curseur se place en fin de ligne physique.

5.8.2.2. Les déplacements ligne

^X Curseur descendu d'une ligne physique.

Le curseur descend d'une ligne tout en restant dans la même colonne.

S'il est en fin de page il y a un décalage de cette dernière ligne écran vers le haut.

Si sur la position colonne il n'y a pas de caractère, le curseur se positionne dans la colonne inférieure la plus proche d'un caractère.

^E

Curseur remonté d'une ligne physique.

Le curseur remonte d'une ligne tout en conservant la même position verticale.

S'il est en début d'écran, il y a un décalage d'une ligne écran vers le haut.

Si sur la position colonne il n'y a pas de caractère, le curseur se positionne dans la colonne inférieure la plus proche d'un caractère.

Ces deux commandes déplacent le curseur d'une ligne écran. Une ligne texte peut être composée d'une ou plusieurs lignes écran : c'est la ligne logique.

5.8.2.3. Les déplacements par mot

Définition d'un mot : un mot est composé de un ou plusieurs caractères. Le premier caractère suit un séparateur de mot, c'est-à-dire tout caractère non alphanumérique (ex : rc, saut de ligne...).

Exemple :

bleu , blanc. La virgule est un mot, blanc en est un autre.

bleu, blanc. Dans ce cas la virgule n'est pas un mot car elle n'est pas précédée d'un espace.

^A

Curseur d'un mot à gauche.

Le curseur se déplace vers la gauche et se place sur le premier caractère du mot précédent. Si le curseur est dans le mot il va d'abord au début du mot.

Si au départ il est en début de ligne, il se met au début du dernier mot de la ligne précédente.

De même s'il est au début de la première ligne de la page, il y a décalage de cette dernière d'une ligne écran vers le bas puis le curseur passe à la ligne précédente.

^F Curseur d'un mot à droite.

Le curseur se déplace vers la droite et se positionne sur le premier caractère du mot suivant. Si au départ il est situé en fin de ligne, il se met au début du premier mot de ligne suivante. Par contre s'il est en fin de la dernière ligne de la page il y a décalage de celle-ci d'une ligne écran vers le haut et apparition de la ligne suivante.

5.8.2.4. Les déplacements par caractère

^S Curseur d'un caractère à gauche.
ou

<- Le curseur se déplace à gauche d'un caractère. Au départ s'il est en début de ligne il se positionne en fin de ligne précédente, par contre s'il est au début de la première ligne de l'écran, il y a décalage d'une ligne écran vers le bas suivi de l'apparition de la ligne précédente.

^D Curseur d'un caractère à droite.
ou
->

Le curseur se déplace d'un caractère vers la droite. Si au départ, il est en fin de ligne, il ira se positionner au début de la ligne suivante, par contre s'il est situé en fin de la dernière ligne écran, il y a décalage de ce dernier d'une ligne écran vers le haut suivi de l'apparition de la ligne suivante.

^Q Curseur d'une tabulation à droite.

Le curseur se déplace d'une tabulation à droite, c'est-à-dire de huit colonnes à droite. Si on arrive en bout de ligne il faut passer à la ligne suivante par un retour chariot.

5.8.3. Les suppressions

L'utilisateur ne doit pas oublier que l'espace, la tabulation et la fin de ligne sont des caractères.

5.8.3.1. Au niveau de la ligne

- ^Y** Suppression d'une ligne logique de texte.
- Pour exécuter cette commande le curseur peut être positionné sous n'importe quel caractère de la ligne. Si une ligne texte se compose de plusieurs lignes écran, ces dernières sont toutes détruites. La place laissée par ces lignes est comblée par les lignes suivantes.
- ^U** Suppression de la ligne logique de texte à gauche.
- L'utilisateur supprime le début de la ligne texte jusqu'au caractère (exclus) sur lequel se trouve le curseur. Le curseur est toujours positionné sur le même caractère.
- ^K** Suppression de la ligne logique de texte à droite.
- L'utilisateur supprime la fin de la ligne texte à partir du caractère (inclus) sur lequel se trouve le curseur. Après opération le curseur ne s'est pas déplacé.

5.8.3.2. Au niveau du mot

- ^_** Suppression de mot à gauche.
- Il y a suppression du début du mot à partir du caractère (exclus) sur lequel se trouve le curseur. Après opération le curseur reste positionné sur celui-ci.
- ^T** Suppression de mot à droite.
- Il y a suppression de la fin du mot à partir du caractère (inclus) sur lequel se trouve le curseur. Ce dernier se positionne, après opération, sur le premier caractère qui suit le mot supprimé.

5.8.3.3. Au niveau du caractère

- Effacement** Suppression de caractère à gauche du curseur.
- La touche effacement de votre clavier supprime le caractère situé à gauche du curseur. Ce dernier reste positionné sur le caractère.
- ^ G** Suppression de caractère repéré par le curseur.
- Le caractère pointé par le curseur est supprimé. Le caractère suivant se positionne sur le curseur.
- Si on refait ^ G après la suppression du dernier caractère de la ligne, la ligne suivante vient se rattacher à ce qui reste de la ligne sur laquelle on a appliqué la commande. Vous avez supprimé le retour chariot qui terminait votre ligne.
- (le retour chariot étant considéré comme un caractère).

5.8.3.4. Commande d'insertion

- ^ O** Insertion de caractères
- Cette commande permet d'insérer des chaînes de caractères, des "retour chariot".
- Le mot insertion apparaît avec un attribut de visualisation sur la ligne d'état.
- Il suffit de refaire ^ O pour quitter le mode insertion.

5.8.3.5. Les autres commandes

- ^ N** Insertion d'un retour chariot
- Cette commande sert à insérer un retour chariot et un saut de ligne à l'emplacement du curseur.
- Cela permet de scinder une ligne en deux si on insère un ^ N.
- Cela permet également de créer des lignes après le curseur.
- M
ou RC** Retour chariot
- Insertion active : le retour chariot est inséré dans le fichier avant le curseur.
 - Insertion non active : curseur au début de la ligne suivante.
 - En fin de fichier : inséré dans le fichier.

^I

Caractère tabulation inséré dans le fichier.

Cette commande est utilisée pour insérer une tabulation de huit blancs à droite à l'emplacement du curseur.

^P

Met le caractère suivant cette commande dans le fichier même si c'est un caractère de contrôle.

Le caractère de contrôle est considéré comme un unique caractère logique : le curseur viendra se positionner sur l'accent circonflexe quand on explore le fichier.

Pour écrire ^O dans le fichier, il faut d'abord faire ^P suivi de ^O. Ainsi ^O sera considéré comme une chaîne de caractères insérés dans le fichier et non considéré comme un caractère de contrôle. En refaisant ^O une deuxième fois, la commande ^O sera exécutée.

Pour insérer un saut de page dans un fichier texte, il faut faire ^P ^L (^L = saut de page).

^J

Affichage de la zone aide utilisateur : les commandes vidéo sont résumées.

ESC

Passage en mode de commande.

^@

Exécute la commande suivante quatre fois.

Deux ^@ consécutifs font exécuter la commande suivante seize fois, trois ^@ consécutifs soixante-quatre fois et quatre ^@ consécutifs deux cent cinquante-six fois. Mais cinq ^@ consécutifs font exécuter la commande suivante quatre fois.

5.8.4. **Présentation et utilisation en mode de fonctionnement commande**

Présentation :

Dans ce mode les commandes sont affichées sur la ligne d'état. Un astérisque est édité et la dernière commande exécutée est affichée à la suite. Le curseur se positionne sous le premier caractère de cette commande.

L'utilisateur a deux possibilités :

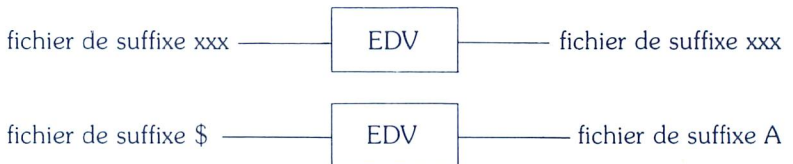
- soit il tape sur la touche retour chariot et l'éditeur exécute une nouvelle fois la même commande.
- soit il tape une nouvelle commande et il la valide par un retour chariot. Celle-ci s'exécute.

Une commande est limitée à 53 caractères.

5.8.4.1. **Commandes de sauvegardes du fichier**

E Conserver les modifications et mettre fin à la mise à jour.

Le programme supprime les fichiers intermédiaires et ferme tous les autres fichiers.



Le programme renomme le fichier source de départ de suffixe XXX en A. Ce fichier ne contient pas les modifications. Le fichier source créé ou modifié (de suffixe \$, celui sur lequel vous travaillez), devient le fichier source de départ de suffixe XXX : il possède les modifications que vous avez effectuées.

Tout ceci quand on n'a pas l'option DEST.

Sinon reportez-vous au schéma récapitulatif de la gestion des fichiers.

H Conserver les modifications, mettre fin à la mise à jour puis revenir en mode vidéo, le curseur positionné au début du fichier.

La sauvegarde des fichiers se fait de façon similaire à la commande précédente.

Mais au lieu de retourner à l'interpréteur de commande de PROLOGUE on revient en mode vidéo.

Q Abandon de la mise à jour

L'éditeur supprime les fichiers intermédiaires et ferme tous les autres fichiers.

Retour à l'interpréteur de commande.

O Retour au fichier source en abandonnant les modifications.

L'éditeur abandonne les mises à jour et recommence la création ou la mise à jour avec le fichier source de départ.

Pour ces deux commandes (Q et O) l'éditeur demande une validation car il y a perte des mises à jour et l'opérateur peut avoir fait une erreur de frappe.

Après la frappe du retour chariot pour l'exécution de la commande EDV pose la question :

"abandon (o/n) ? :"

Si l'opérateur répond par la négative il n'y a pas de destruction des modifications et l'éditeur se met en attente de la frappe d'une nouvelle commande. Dans le cas contraire il y a abandon des modifications et le programme retourne en début de fichier.

^Q Affichage ou suppression de la zone aide utilisateur.

Caractères spéciaux pour la saisie d'une commande :

Flèche avant : déplace le curseur d'un caractère vers la droite s'il n'est pas en fin de commande.

Flèche arrière : déplace le curseur d'un caractère vers la gauche s'il n'est pas en début de commande.

Effacement ou escape : efface le caractère situé à gauche du curseur.

^A Positionne le curseur en début de commande.

5.8.4.2. Commandes diverses

Caractères spéciaux appartenant à la syntaxe des commandes :

— Symboles particuliers dans la syntaxe :

+ - indique le sens d'exécution d'une commande.
Par défaut le signe + est pris en compte.

n nombre appliqué à une commande qui vaut un par défaut. Si on met un # à la place, le nombre pris en compte vaut 65535.

— Caractères spéciaux :

^N est converti en retour chariot suivi d'un saut de ligne.

^Y est converti en un escape.

Ces caractères évitent d'employer les touches "ESCAPE" (sortie) et RETOUR CHARIOT pour éviter toute confusion dans un libellé "texte", notamment lors d'une recherche de caractères.

Si l'argument de la commande comporte soit un code "ESCAPE" soit un code RETOUR CHARIOT, on utilise respectivement soit ^Y soit ^N.

De plus pour les libellés “arguments” :

- ^B** permet de repérer n'importe quel caractère.
- ^T** permet de repérer n'importe quel séparateur.
- ^Ox** repère tout caractère autre que le caractère “x”. Mais ^Ox sera détecté s'il est présent dans le fichier.

Pour les libellés “argument” ces caractères spéciaux sont en plus de ceux pour les libellés “texte”.

En mode commande, ^Z et ^J sont interdits.

5.8.4.3. Les commandes de déplacement du pointeur et du curseur et d'affichage

V Retour en mode vidéo.

Il permet de visualiser le fichier.

+B ou -B Positionnement du curseur et du pointeur en début (+) ou en fin (-) de fichier.

L'utilisateur positionne à l'aide de cette commande le pointeur en début ou fin de fichier. Si le signe n'est pas précisé, l'éditeur prend en implicite le signe +.

B signifie begin : commencer.

Ainsi le pointeur va se positionner sur le premier caractère du fichier.

Pour mieux comprendre le sens du signe, il faut l'interpréter comme une volonté de l'opérateur d'explorer le fichier dans le sens positif (début/fin) ou négatif (fin/début).

+ -nC Déplace le pointeur de n caractères.

Le retour chariot saut de ligne est assimilé à un caractère.

+ -nL Déplace le curseur de n lignes.

En fin de commande, le curseur est positionné en début de ligne.

+—nT	Affiche n lignes. Le pointeur ne change pas de place. La commande OT n'a aucun effet.
+—nG	Déplace le pointeur de n lignes suivi de l'affichage de la ligne. En fin de commande le pointeur est situé au début de la ligne. La commande OG permet de visualiser la ligne sur laquelle se trouve le pointeur.
+—nP	Déplace le pointeur d'une page et affiche n pages. En fin de commande le pointeur est situé au début de la première page. Une page est équivalente à la zone d'affichage que constitue une page écran. La commande OP affiche la page où se trouve le pointeur. Celui-ci n'est en aucun cas déplacé.

5.8.4.4. Les commandes de suppression de texte

+—nD	Efface n caractères. Le retour chariot saut de ligne est assimilé à un caractère. Le pointeur est situé sur le premier caractère non effacé.
+—nK	Efface n lignes. Le pointeur est situé au début de la première ligne non effacée.

5.8.4.5. Les commandes d'insertion

n!texte	Insère n fois "texte" à partir de la position du pointeur de caractère.
nAtexte	Exécute n fois la commande suivante : déplacement du pointeur d'une ligne suivi de l'insertion de "texte".
I	Passes en mode vidéo avec l'insertion effective.
A	Positionne le pointeur sur la ligne suivante puis passe en mode vidéo avec insertion effective.
n!	Insère dans le fichier un caractère dont le code ASCII est n, n variant de 0 à 255.

5.8.4.6. Commandes de recherche de chaînes de caractères

—+nNch.caract. Recherche d'une chaîne de caractères vers la fin (+) ou le début (—) du fichier.

L'utilisateur recherche une chaîne de caractères à l'intérieur de son fichier.

La frappe de caractères dans la commande doit exactement correspondre à celle de la chaîne recherchée, notamment au niveau des majuscules/minuscules.

n est un nombre qui vaut un par défaut, et au maximum 65535 que l'on peut remplacer par # dans la commande pour éviter d'écrire ce nombre. Il indique la n^{ème} chaîne recherchée après la position du pointeur dans le fichier dont l'image à l'écran est le curseur.

Il faut toujours revenir en mode vidéo pour mettre à jour la position du curseur par rapport au pointeur.

La longueur de cette chaîne est limitée par celle de la zone allouée à la commande.

La recherche se faisant vers le début ou la fin de fichier, EDV prendra en implicite la fin de fichier. Quand la chaîne est trouvée le pointeur est positionné sur la chaîne dans le fichier et à l'écran le curseur se positionne sur celle-ci si l'on revient en mode vidéo.

Le curseur est l'affichage d'une position sur l'écran, tandis que le pointeur est la position réelle de travail dans le fichier.

En mode vidéo le curseur est l'image exacte de la position du pointeur dans le fichier.

Par contre en mode commande cela n'est pas toujours vrai.

Cette commande peut être interrompue à tout moment en enfonçant la touche ESCAPE.

EDV revient en mode commande.

Pour affiner la recherche, on peut mettre ^N après N pour la localiser en début de ligne.

Ceci est valable pour les autres commandes.

Pour les commandes qui suivent, n a la même fonction que précédemment.

+—nFchai.caract. Recherche d'une chaîne de caractères vers la fin (+) ou le début (—) de la partie du fichier présente en mémoire centrale.

Si le signe n'est pas précisé, le + est implicite.

+—nSchain.caract.ESCnouv.chaine Recherche d'une chaîne de caractères vers la fin ou le début du fichier en zone mémoire puis substitution de celle-ci par la nouvelle chaîne. La longueur des deux chaînes n'a pas d'importance, l'éditeur se chargeant du recadrage à l'intérieur du fichier.

+—nRchain.caract.ECSnouv.chaine Recherche d'une chaîne vers le début ou la fin du fichier puis substitution de celle-ci par la nouvelle chaîne. La longueur des chaînes et le recadrage sont identiques à la commande précédente.

Toutes les commandes de recherches peuvent être interrompues à tout moment par l'enfoncement de la touche ESCAPE.

Ces commandes de substitution permettent la suppression d'un texte si nouv.chaîne est omis.

Exemple :Schain.caract.ESC <rc>

Un message d'erreur est émis si chain.caract. n'est pas trouvée.

Si on met un / avant F ou N ou S ou R, la commande ne provoquera pas l'affichage d'un message d'erreur. Le curseur viendra se positionner devant l'étoile.

5.8.4.7. Zone de travail pour manipuler des parties de fichier

Vous disposez également d'une zone de travail pour manipuler des lignes du fichier :

nQP

Placer n lignes dans la mémoire de travail et supprimer ces lignes du fichier.

La première ligne commence à partir du pointeur positionné dans le fichier.

A la fin de cette commande, le pointeur est positionné sur le premier caractère de la ligne située juste après la suppression.

Si l'on revient en mode vidéo, le curseur sera l'image de la position du pointeur.

n/QP

Ajouter n lignes dans la mémoire de travail et supprimer ces lignes du fichier.

La première ligne commence à partir du pointeur positionné dans le fichier.

Ces lignes viennent s'ajouter à la suite de celles déjà présentes dans la mémoire de travail.

A la fin de cette commande le pointeur est situé sur le premier caractère de la ligne qui suit la suppression.

nQG	Copier la mémoire de travail dans le fichier n fois. Les lignes de la mémoire de travail sont insérées dans le fichier à partir de la position du pointeur dans celui-ci. N'oubliez pas de revenir en mode vidéo pour voir le résultat de ces commandes au niveau du fichier.
QT	Visualisation du contenu de la mémoire de travail.
QK	Vider la mémoire de travail de son contenu.
nQP\$	Place n lignes dans la mémoire de travail sans supprimer ces lignes du fichier. Cette commande a la même fonction que nQP mais les lignes ne sont pas supprimées dans le fichier. Le pointeur ne change pas de place.
n/QP\$	Cette commande est similaire à la précédente. Mais ces n lignes sont insérées dans la mémoire de travail à la suite de celles déjà présentes.
nQLtexte	Place n fois "texte" dans la mémoire de travail "texte".
n/QLtexte	Ajoute n fois "texte", à la suite du texte déjà présent dans la mémoire de travail.
nQX	Exécute n fois le texte de la mémoire de travail qui est constitué par une commande ou une suite de commandes chaînées. Le texte de la mémoire de travail peut être introduit par l'intermédiaire des commandes QL ou QP. La longueur du texte n'est limitée que par la capacité de la mémoire. Le texte peut contenir des retours-chariots sauts de ligne (^N). Il ne faut pas séparer les commandes par un retour chariot. Lors de l'emploi de la commande QL, il est possible d'insérer des "ESCAPE" en tapant ^Y.

Remarque :

L'enchaînement des commandes est réalisable.

Il y a deux sortes de commandes, d'une part les commandes sans paramètre comme B et V, et celles avec paramètre comme N.

On ne met pas de séparateur entre :

- deux commandes sans paramètre
- une commande sans paramètre suivie d'une autre avec paramètre.

Par contre le séparateur est nécessaire :

- entre une commande avec paramètre suivie d'une commande avec o'' sans paramètre.

Le séparateur est : ^[

Il suffit d'enfoncer la touche ESCAPE pour l'obtenir sur la ligne d'état.

Exemple d'enchaînement de commandes :

```
-----  
: fichier : EXEMPLE—S          *bv                                     :  
-----  
:                                                                    :  
:                                                                    :  
:                                                                    :  
-----
```

L'utilisateur demande un positionnement du pointeur en début de fichier suivi d'un passage en mode vidéo.

Ceci est une suite de deux commandes sans paramètres.

```
-----  
: fichier : EXEMPLE—S          *bnétiquette ^ [v                       :  
-----  
:                                                                    :  
:                                                                    :  
:                                                                    :  
-----
```

L'utilisateur demande un positionnement du pointeur en début du fichier suivi d'une recherche de chaîne suivie d'un passage en mode vidéo.

Comme l'on revient en mode vidéo, le curseur sera l'image du pointeur.

Les commandes E, O, H, Q et V arrêtent l'enchaînement des commandes. Pour E et Q, EDV redonne la main à l'interpréteur de commandes. Par contre pour les commandes O, H, V, EDV retourne au mode vidéo. Dans les deux cas les commandes suivant E, O, H, Q, V, ne sont pas prises en compte.

Par exemple : *Ev

Seul E est pris en compte.

5.8.4.8. Mouvements de fichiers

Ces commandes sont surtout utilisées pour la fusion de fichiers, le partage de l'un d'entre eux ou le transfert de textes trop importants pour être manipulés par l'intermédiaire de la mémoire de travail.

Y[unité.]nomfichier-type Insère le fichier à l'endroit où se trouve le pointeur.

Le fichier stipulé est lu et inséré dans le fichier édité au niveau du pointeur de caractère. Cette opération terminée, le pointeur est positionné après le dernier caractère du fichier lu.

nW[unité.]nomfichier-type Ecrire les n lignes suivant le pointeur de caractère dans le fichier stipulé.

Le pointeur se positionne après le dernier caractère écrit. Le fichier stipulé doit être inexistant sur le support.

Il n'existe aucun moyen permettant d'insérer ces lignes en fin de fichier.

5.8.4.9. Les boucles

n<ligne de commande> Répéter n fois la ligne de commande située
ou à l'intérieur des <>

nMligne de commande>

Si n est omis, 65535 est pris par défaut. Ceci permet donc de terminer la boucle sans incident.

Exemple :

B<Nveston(ESC)V>

Cette commande recherche tous les “veston” du fichier.

A chaque fois qu’un “veston” est détecté, on passe en mode vidéo et le pointeur est positionné juste après.

L’opérateur peut alors agir à sa guise sur le texte situé autour de “veston”, puis il suffit d’appuyer sur “ESCAPE” pour quitter le mode vidéo. EDV cherchera alors le prochain “veston”, passe en mode vidéo. Cette opération se répétera tant que la dernière chaîne de caractères “veston” ne sera pas trouvée.

5.9.

Enfoncez la touche ESCAPE pour passer en mode commande.

- représente le curseur.

```

: fichier : EXEMPLES—S      *
-----
:      TOTO2
:      TOTO3
:      TOTO4

```

En validant la commande `bv`, `b` vous place en début de fichier. Cette opération est visualisée par `v` : vous êtes revenu en mode vidéo.

```

: fichier : EXEMPLES—S      *bv
-----
:  dynastie des TOTQ
:
:      TOTO1
:      TOTO2
:      TOTO3
:      TOTO4

```

La commande est identique, mais vous allez en fin de fichier car il y a un signe moins.

```

-----
: fichier : EXEMPLES—S      *—bv
:
:  dynastie des TOTO
:
:      TOTO1
:      TOTO2
:      TOTO3
:      TOTO4
:

```

Recherche de chaînes de caractères :
On veut retrouver la chaîne “tra” dans un texte.

```

-----
: fichier : TOMDOM—S      *—2Ntra ^ [v
:
: L'ordinateur sert au travail.
: Le travail c'est la santé.
: Bien que le repos soit également la santé,
: L'informatique ne fait pas (encore) tout notre travail
: Mais elle travaille quand même à notre santé.
:

```

La commande `-2Ntra$ v<rc>` entraîne la recherche de la deuxième chaîne “tra” en allant vers le début du fichier. Puis vous revenez en mode vidéo.

En enfonçant la touche ESCAPE (`^ [`) (`$`), vous repassez en mode commande et la commande précédente réapparaît.

En faisant un retour chariot pour la revalider, EDV va rechercher la deuxième chaîne “tra” qui précède celle qu'il vient de trouver.

```

-----
: fichier : TOMDOM—S      *—2Ntra ^ [v
:
: L'ordinateur sert au travail.
: Le travail c'est la santé.
: Bien que le repos soit également la santé,
: L'informatique ne fait pas (encore) tout notre travail
: Mais elle travaille quand même à notre santé.
:

```


6.	LES EXTENSIONS DE PROLOGUE ET LEURS COMMANDES	6-1
6.1.	Généralités	6-1
6.2.	Les décors	6-2
6.2.1.	Généralités	6-2
6.2.1.1.	Principes de fonctionnement	6-3
6.2.1.2.	Conversion des fichiers natifs	6-4
6.2.1.3.	Transparence PROLOGUE	6-4
6.2.1.4.	Utilisation des caractères nationaux	6-5
6.2.1.5.	Partage des fichiers en configuration multipostes	6-6
6.2.1.6.	Précautions d'utilisation des DECORS	6-9
6.2.2.	Le décor CP/M-86	6-10
6.2.2.1.	La commande CPM	6-10
6.2.2.2.	La commande LCPM— : consultation du répertoire	6-12
6.2.2.3.	La commande LCPM— : transfert des fichiers	6-13
6.2.3.	Le décor MS-DOS	6-15
6.2.3.1.	La commande MSDOS	6-15
6.2.3.2.	La commande LMS— : consultation du répertoire	6-16
6.2.3.3.	La commande LMS— : transfert des fichiers	6-17
6.3.	Le séquentiel indexé	6-19
6.3.1.	Généralités	6-19
6.3.2.	Sauvegarde des fichiers S.I. : "SVSI"	6-20
6.3.3.	Restauration des fichiers S.I. : "RTSI"	6-24
6.4.	Le multi-critères	6-28
6.4.1.	Généralités	6-28
6.4.2.	Sauvegarde des fichiers multi-critères	6-28
6.4.3.	Restauration des fichiers multi-critères	6-29
6.5.	La base de données	6-30
6.5.1.	Généralités	6-30
6.5.2.	Sauvegarde d'une base de données	6-30
6.5.3.	Restauration des bases de données	6-31

6.6.	Mise en œuvre du B.A.L.	6-32
6.6.1.	L'exécuteur "EX"	6-32
6.7.	Utilitaires B.A.L.	6-36
6.7.1.	Inhibition des granules "TOPGR-T"	6-36
6.7.2.	Programmation des touches fonctions : "TFP"	6-38
6.8.	Le graphique	6-45
6.8.1.	Généralités	6-45
6.8.2.	Description des dessins	6-47
6.8.2.1.	Espace de travail/Espace de visualisation	6-47
6.8.2.1.1.	espace travail	6-47
6.8.2.1.2.	Espace de visualisation	6-47
6.8.2.2.	Fenêtre/Clôture	6-48
6.8.2.3.	Clôture réelle	6-51
6.8.3.	Fonctionnalités du graphique	6-53
6.8.3.1.	Assignation d'un numéro logique	6-53
6.8.3.2.	Libération d'un numéro logique	6-54
6.8.3.3.	Définition de fenêtre et de clôture	6-54
6.8.3.4.	Ordres de tracé	6-58
6.8.3.4.1.	Tracé de symboles	6-58
6.8.3.4.2.	Tracé de lignes	6-59
6.8.3.5.	Attributs de tracé	6-60
6.8.3.5.1.	Attributs de couleur	6-60
6.8.3.5.2.	Attributs de type	6-61
6.8.3.5.2.1.	Type de symbole	6-61
6.8.3.5.2.2.	Type de ligne	6-61
6.8.3.5.3.	Attributs de taille	6-62
6.8.3.5.3.1.	Taille de symbole	6-62
6.8.3.5.3.2.	Epaisseur de ligne	6-62
6.8.3.6.	Effacement de clôture	6-63
6.8.3.7.	Lecture des paramètres associés à un numéro logique	6-63
6.8.3.8.	Lecture de l'attribut couleur d'un point écran	6-64

6.8.4.	Fonctionnement du système graphique	6-65
6.8.4.1.	Organisation système	6-65
6.8.4.2.	Portabilité	6-65
6.8.4.3.	Niveau minimum de fonctionnalités graphiques	6-66
6.8.4.4.	Principe de fonctionnement	6-66
6.9.	Les transmissions	6-68
6.9.1.	La procédure PTTY	6-68
6.9.1.1.	Introduction	6-68
6.9.1.2.	La configuration de la procédure PTTY	6-69
6.9.1.3.	La commande de transmission de fichiers : "TELE"	6-70
6.9.2.	La procédure B.S.C. 2780-3780-3741	6-75
6.9.2.1.	Introduction	6-75
6.9.2.2.	La configuration de la procédure B.S.C. 2780-3780-3741	6-77
6.9.3.	La procédure B.S.C. 3270	6-79
6.9.3.1.	Introduction	6-79
6.9.3.2.	La configuration de la procédure B.S.C. 3270	6-80
6.9.4.	La procédure VIP/PVE	6-83
6.9.4.1.	Introduction	6-83
6.9.4.2.	La configuration de la procédure VIP/PVE	6-84
6.9.5.	La procédure X25	6-85
6.9.5.1.	Introduction	6-85
6.9.5.2.	La configuration de la procédure X25	6-86
6.10.	Les réseaux	6-87
6.10.1.	Généralités	6-87
6.10.2.	Le réseau SPR	6-88
6.10.2.1.	Introduction	6-89
6.10.2.2.	La configuration du réseau SPR	6-91

6.11.	L'option "MULTIVUE"	6-92
6.11.1.	Introduction	6-92
6.11.2.	Fonctionnement	6-94
6.11.2.1.	Mode général	6-94
6.11.2.2.	Mode graphique	6-94
6.11.2.3.	Mode transparent	6-94
6.11.3.	L'utilitaire "POSTE"	6-95
6.11.3.1.	L'allocation ou la commutation	6-95
6.11.3.2.	La désallocation	6-96
6.11.3.3.	L'envoi de caractères	6-96
6.11.4.	Présentation de la 25 ^{eme} ligne système	6-97

6. LES EXTENSIONS DE PROLOGUE ET LEURS COMMANDES

6.1. GENERALITES

Les extensions de PROLOGUE sont des logiciels associés au système d'exploitation et qui étendent les fonctionnalités de celui-ci. Ces options sont accédées au moyen de programmes, non décrits dans le chapitre 4 parce que ne faisant pas partie du système de base. Suivant la configuration de PROLOGUE que vous utilisez, il est possible que ces programmes soient ou non présents sur les volumes utilisés.

6.2. LES DECORS

6.2.1. Généralités

Ce chapitre est consacré à un aspect original de PROLOGUE que l'on appelle les fonctions « DECORS ». Un DECOR est un programme utilitaire qui permet d'exécuter sous PROLOGUE des logiciels et des progiciels conçus pour être mis en œuvre avec d'autres systèmes d'exploitation de micro-ordinateurs.

L'emprunt du mot « décor » au vocabulaire du théâtre pour désigner cette fonctionnalité s'explique par le fait que PROLOGUE plante le décor du système d'exploitation concerné et exécute les programmes que celui-ci supporte habituellement comme si ce dernier était réellement présent en mémoire.

Il existe actuellement deux utilitaires DECORS : CPM-X et MSDOS-X avec lesquels PROLOGUE peut émuler respectivement les systèmes d'exploitation suivants :

CP/M 86 (R) et MS.DOS (R)

Grâce à la mise en œuvre de ce concept, PROLOGUE vous offre la possibilité d'exécuter sur votre micro-ordinateur un grand nombre de programmes d'application issus de l'environnement de CP/M 86 et de MS.DOS, sans que le système natif ne soit présent.

En environnement multiposte, on peut voir tourner simultanément des programmes d'origine PROLOGUE, CP/M 86 et MS.DOS.

(R) DECOR est une marque déposée de BULL MICRAL

(R) CP/M 86 est une marque déposée de DIGITAL RESEARCH

(R) MS.DOS est une marque déposée de MICRO SOFT

6.2.1.1. Principes de fonctionnement

Le signal d'appel de PROLOGUE : —> est remplacé par celui du système simulé par le DECOR : A>. Une commande que vous entrez avec la syntaxe du système simulé sera interprétée par le DECOR et exécutée par PROLOGUE.

La ligne de commande ne doit comporter qu'une des principales fonctionnalités standards du système natif simulé par le DECOR correspondant.

L'édition des résultats est faite selon les normes habituelles de la fonction simulée.

DECOR reconnaît la structure des noms de fichier de CP/M 86 et MS.DOS.

Les fonctionnalités de la plupart des utilitaires « résidents » ou « internes » de CP/M 86 et MS.DOS sont actives sous DECOR. Par contre, les utilitaires ayant leurs équivalents sous PROLOGUE, ou dépendant de caractéristiques matériels spécifiques, ne sont pas pris en compte par DECOR.

Toutefois certains utilitaires externes de ces systèmes comme EDLIN (MD.DOS), ED (CP/M 86), DEBUG (MS.DOS), DDT (CP/M 86) fonctionnent sous DECOR, après conversion du fichier correspondant.

Si la commande consiste dans le lancement de l'exécution d'un programme d'application ou d'un progiciel, DECOR gère l'appel aux ressources « système » de PROLOGUE nécessaires, en particulier pour la gestion des fichiers.

Attention :

Les programmes DECORS ne sont pas des émulateurs.

Les fichiers des utilitaires CP/M 86 et MS.DOS sont fournis avec la disquette de distribution de PROLOGUE.

6.2.1.2. Conversion des fichiers natifs

Un décor est un utilitaire PROLOGUE. Il ne peut travailler qu'avec des supports magnétiques sectorisés au format de PROLOGUE. Il ne gère pas les supports au format du système natif.

Sectorisation des disquettes 5" 1/4	
PROLOGUE	16 secteurs de 256 octets
CP/M 86	16 secteurs de 128 octets
MS.DOS	8 ou 9 secteurs de 512 octets

6.2.1.3. Transparence PROLOGUE

Pour ne pas pénaliser les utilisateurs de PROLOGUE, les programmes conçus sous d'autres systèmes d'exploitation peuvent être lancés directement à partir de l'interpréteur de commande de PROLOGUE. L'interpréteur reconnaît le système d'exploitation d'origine du fichier exécutable à partir de son type.

L'interpréteur de PROLOGUE charge automatiquement le DECOR correspondant qui prend alors le contrôle du programme. Ceci est totalement transparent pour l'utilisateur. La syntaxe est celle de PROLOGUE.

Suffixes des Fichiers	
PROLOGUE	X ou T
CP/M 86	CMD
MS.DOS	COM ou EXE

Attention :

Il n'est pas possible d'utiliser les mnémoniques PROLOGUE des périphériques dans une commande destinée à être exécutée par une fonction DECOR.

Exemples :

```
->BASIC,DEMO <rc>  
->ASM,SOURCE <rc>  
->BASCOM B:T BASIC.BAS <rc>  
->LINK, TMSM TOTO TITI,TOTO,NUL:,TATA <rc>
```

Inversement, il est toujours possible de frapper une commande PROLOGUE après le signe d'appel d'un DECOR. A la fin de l'exécution de la commande système reste au niveau de l'interpréteur DECOR.

6.2.1.4. Utilisation des caractères nationaux

Un programme DECOR est un utilitaire de PROLOGUE, et à ce titre, il reconnaît les caractères standard de PROLOGUE (utilisation du huitième bit). Ainsi un programme d'application conçu dans ce contexte peut accéder à tous les caractères nationaux.

Cette flexibilité rend moins aisée l'exécution de certains logiciels anglo-saxons d'édition de ligne ou de texte, limités aux caractères ASCII (codes sur 7 bits) ou encore utilisant le huitième bit pour leur gestion interne.

Les DECORS, pour pallier à cette difficulté et pour permettre à ces logiciels d'utiliser des caractères nationaux, ont adopté la méthode de substitution de codes. Ainsi les caractères ASCII peu utilisés en traitement de texte voient leur code affecté à des caractères nationaux.

Table de substitution Franco-Belge standard							
à	é	è	ù	ç	§	°	· ·
@	#	]	l	\	[	\	~

L'utilisateur peut choisir dans ce cas un des deux modes de fonctionnement suivants :

- ASCII standard (sans caractères accentués)
- Substitution de code ASCII (caractères nationaux)

Chaque mode est exclusif de l'autre. Au chargement du logiciel, c'est le mode ASCII standard qui est retenu (mode implicite). Le passage à l'autre mode est obtenu par la frappe simultanée de CTRL-@. Cette commande est alternative.

Remarque :

Le codage des caractères accentués sous DECOR étant différent dans ces cas de celui de PROLOGUE, il est indispensable qu'un texte saisi sous DECOR soit édité sous DECOR. L'édition de ce texte directement avec PROLOGUE donnerait des graphismes différents de ceux choisis.

6.2.1.5. Partage des fichiers en configuration multiposte

MS-DOS et CP/M-86 sont des systèmes d'exploitation monotâches et monopostes.

Il n'y existe donc pas de primitives système qui permettent le partage des fichiers.

Sous décor CP/M-86 ou MS-DOS, les fichiers sont ouverts en mode exclusif. Ce qui peut empêcher les programmes CP/M-86 ou MS-DOS de fonctionner sur PROLOGUE multiposte.

Des aménagements ont été faits au niveau du décor pour limiter ces inconvénients :

- Les fichiers de commandes sont ouverts en mode partageable, ce qui permet de faire fonctionner des programmes monofichier (qui n'ont pas de fichiers intermédiaires ou overlays), sur plusieurs postes.
- Les fichiers de données sont ouverts en mode exclusif.

Exemple :

Sur le poste 0, l'utilisateur utilise la commande :

—> WM,SOURCE-TXT

Le fichier "WM-CMD"	est ouvert en mode partageable
Le fichier "source-TXT"	est ouvert en mode exclusif

Sur le poste 1,

—> WM,AUTRE-TXT La commande est correcte

—> WM,SOURCE-TXT Erreur : fichier SOURCE-TXT
non partageable

La commande est incorrecte car le fichier SOURCE.TXT n'est pas partageable ; le décor signale l'erreur 44 (fichier non partageable).

Cette solution ne peut résoudre le problème de partage de fichier de certains programmes ayant des fichiers overlays (Help de Wordmaster, Supercalc, ...).

Pour permettre de résoudre une partie de ce problème, une table des fichiers à partager a été introduite au niveau du décor. Quand l'ouverture d'un fichier est demandée, le décor consulte cette table ; si le suffixe du fichier se trouve dans la table, il sera ouvert en mode partageable. Ainsi il permet à un certain nombre de programmes de fonctionner simultanément sur plusieurs poste (commande HELP de WM, Word Star, etc ...).

Exemple :

Si le suffixe HLP se trouve dans la table, le fichier WM-HLP sera ouvert en mode partageable ; on peut donc l'utiliser sur plusieurs postes.

Le début de la table des fichiers partageables se trouve au secteur 3 du fichier CPM-X ou MSDOS-X, octet 28H. Il y a 16 entrées, chaque entrée ayant 3 octets définissant le suffixe des fichiers à partager.

Cette table peut être modifiée (ajout ou suppression) par l'utilitaire PATCH-X de PROLOGUE, par la commande :

—> patch,cpm-x,nsec=3

ou

—> patch,MSDOS-X,nsec = 3

Unité : 0	Fichier : CPM-X	Secteur : 3	
00:00200000	00000000	00000000	00000000
10:00000000	00000000	00000000	00000000
20:00000000	00000000	00000000	00000000
30:50444154	5359534C	4F44424F	4D4C4942
40:20202020	20202020	20202020	20202020
50:20202020	20202020	00000000	00000000
60:00000000	00000000	00000000	00000000
70:00000000	00000000	00000000	00000000
80:00000000	00000000	00000000	00000000
90:00000000	00000000	00000000	00000000
A0:00000000	00000000	00000000	00000000
B0:00000000	00000000	00000000	00000000
C0:00000000	00000000	00000000	00000000
D0:00000000	00000000	00000000	00000000
E0:00000000	00000000	00000000	00000000
F0:00000000	00000000	00000000	00000000

.....OVROVLHL
PDATSYSLODCOMLIB

Dans cet exemple, les fichiers de suffixe :

— OVR - OVL - HLP - DAT - SYS — LOD - COM — LIB

sont partageables.

ATTENTION :

La partageabilité est effectuée en lecture comme écriture.
Il faut faire très attention dans la définition des fichiers partageables. Pour ne pas avoir des problèmes d'écrasement de fichier, il est indispensable que seuls les fichiers utilisés en lecture soient partageables.

Contre-exemple CP/M-86 :

Le PASCAL MT⁺ et LINKMT créent des fichiers temporaires de même nom et de même suffixe (PASTEMP-TOK, COMNDATE-\$\$\$, ...) et les utilisent en écriture. Ces fichiers temporaires ne peuvent donc pas être déclarés partageables. Ainsi, le PASCAL MT⁺ et LINKMT ne peuvent pas être utilisés en multiposte, simultanément.

Contre-exemple MS-DOS :

PASCAL, FORTRAN et COBOL de MICROSOFT créent des fichiers temporaires de même nom et même type (PASIBF-BIN,PASIBF-SYM,...) et les utilisent en écriture. Ces fichiers temporaires ne doivent donc pas être partagés.

Ainsi, ces logiciels ne peuvent pas être utilisés simultanément sur plusieurs postes.

6.2.1.6. Précautions d'utilisation des décors

Tous les programmes exécutables sous leur système natif ne fonctionnent pas sous DECOR, s'ils font appel aux spécifications matérielles d'un micro-ordinateur particulier. Il faut rappeler que ce programme ne fonctionnerait pas avec le système natif si l'on passait sur une autre machine.

Les programmes utilisant un accès direct au répertoire ne fonctionnent pas sous DECOR, car la structure du catalogue de PROLOGUE est différente de celle des répertoires des systèmes natifs (CP/M 86 ou MS-DOS).

Certains logiciels MS-DOS étant conçus pour un environnement monoposte (modification directe des relais d'interruption des primitives, libération de mémoire en cours de fonctionnement...), ne peuvent également fonctionner qu'en monoposte sous DECOR.

6.2.2. Décor CP/M 86

6.2.2.1. La commande CPM

Objet de la commande :

La commande CPM charge en mémoire le programme DECOR CP/M-86, interpréteur émulé de CP/M 86. Dès la fin du chargement le signal d'appel de CP/M 86 :

A > apparaît à l'écran. Vous êtes alors placés dans les conditions d'exploitation caractéristiques de CP/M 86, ce qui signifie que vous êtes en mesure d'exécuter les commandes CP/M 86, comme si vous étiez sous le contrôle de ce système d'exploitation.

Syntaxe :

—> CPM <rc>

La bannière du DECOR, la correspondance entre les unités de CP/M 86 et celles de PROLOGUE et l'unité par défaut sont affichées sur l'écran.

tm
DECOR CP/M 86 Version 1.2a
Copyright (c) 1983 BULL Micral
CP/M 86/Prologue Units Equals :

*A: < = > FL0

*B: < = > FL1

A >

Le déroulement des opérations :

Dès l'apparition du signal d'appel A>, vous vous trouvez dans un environnement CP/M 86 ; vous pouvez alors procéder à l'exécution, dans la syntaxe CP/M 86, des commandes résidentes, à l'exception de USER.

Le décor CP/M 86 possède actuellement quatre commandes internes se trouvant en mémoire :

- DIR : Consultation du catalogue des supports disque.
- ERA : Suppression d'un fichier ou d'un groupe de fichiers.
- REN : Renommer un fichier.
- TYPE : Lister un fichier source.

Les caractères séparateurs peuvent être choisis indifféremment selon les règles propres à CP/M 86 ou à PROLOGUE.

Attention :

- Les fichiers R/O et SYS ne sont pas gérés.
- Il n'est pas nécessaire de frapper « Contrôle-C » (CTRL-C) après un changement de disque ou de disquette.
- Les sous-catalogues ne sont pas gérés
- Les bits USER ne sont pas gérés
- L'écriture avec trous n'est pas gérée.

Exemple :

Les quatre commandes :

```
A > WM, PROG01.ASM <rc>
A > WM, PROG01-ASM <rc>
A > WM PROG01.ASM <rc>
A > WM PROG01-ASM <rc>
```

ont des effets identiques.

Le retour au système d'exploitation PROLOGUE s'effectue en appuyant sur la touche « Escape » (ESC).

Notez que vous pouvez aussi exécuter des commandes CP/M non résidentes, directement à partir de l'interpréteur de commande PROLOGUE (->). Dans ce cas, PROLOGUE charge, de manière transparente, le programme CPM-X, puis l'utilitaire non résident.

Exemple :

```
-> FL0.BASIC <rc>
```


6.2.2.2. La commande LCPM-- : consultation du répertoire

Objet de la commande :

La commande LCPM-- vous permet de consulter sous PROLOGUE le répertoire d'un support de format CP/M natif. Elle permet également de transférer un fichier d'un support formaté sous CP/M natif sur support de format PROLOGUE.

Syntaxe 1 :

-> [Unité.]LCPM--,DIR,Unité1 <rc>

où :

Unité mnémonique de l'unité où se trouve le programme utilitaire LCPM--. L'unité implicite système est prise par défaut.

Unité1 mnémonique de l'unité où se trouve la disquette de format CP/M dont vous voulez consulter le répertoire.

Attention :

LCPM-- est un utilitaire de PROLOGUE. Vous devez respecter la syntaxe des commandes et mnémoniques de PROLOGUE.

Exemple :

-> FL0.LCPM--,DIR,FL1 <rc>

A > DIR A:

SYSTEME		:SYSCONF	X	:SYSERF	S
:SYST00	C	:STSCONF	S	:/	X
:WSOVLY	OVR	:WSMSGs	OVR	:WM	CMD
:WM	HLP	:CPM	X	:DECOR	BAX
:SPLO1	X	:LO50	X		

6.2.2.3. La commande LCPM-- : Transfert des fichiers

Objet de la commande :

La deuxième fonction de la commande LCPM-- est de transférer un fichier d'un support de format CP/M 86 sur un support de format PROLOGUE.

Syntaxe 2 :

-> [Unité.]LCPM--,[Unité1.]nomfich-CPM,Unité2 <rc>

ou

-> [Unité.]LCPM--,[Unité1.]nomfich-CPM,[Unité2.]nomfi-PLG <rc>

où :

unité mnémonique de l'unité où se trouve le programme utilitaire LCPM--. L'unité implicite système est prise par défaut.

unité1 mnémonique PROLOGUE de l'unité où se trouve le support de format CP/M.

nomfich-CPM nom du fichier CP/M à transférer sur un support PROLOGUE. Le séparateur peut être indifféremment celui de CP/M ou celui de PROLOGUE (soit le point, soit le tiret).

unité2 mnémonique PROLOGUE de l'unité où se trouve le support de format PROLOGUE qui reçoit le fichier transféré.

Si seule la mnémonique « unité2 » est présente dans la ligne de commande, le fichier sur le support PROLOGUE de destination aura le même nom que sur le support de format CP/M.

nomfi-PLG nom attribué au fichier transféré sur le support PROLOGUE. Ce nom peut être différent de celui du fichier sur le support de format CP/M.

Remarque :

La taille maximale des fichiers pouvant être transférés est de 512K octets.

Exemple :

```
-> MD0.LCPM--,PIP-CMD,MD0 <rc>  
-> MD0.LCPM--,FL0.DDT-CMD,MD0 <rc>  
-> FL1.LCPM--,FL0.TOTO-ASM,FL1.TOTO-ASM <rc>  
-> FL0.LCPM--,FL1.TUTU-BAS,TATA-BAS <rc>
```

6.2.3. Décor MSDOS

6.2.3.1. La commande MSDOS

Objet de la commande :

La commande MSDOS charge en mémoire le programme Décor MS.DOS, interpréteur émulé de MS-DOS. Dès la fin du chargement le signal d'appel de MS.DOS : A >, apparaît à l'écran. Vous êtes alors placés dans les conditions d'exploitation caractéristiques de MS.DOS, ce qui signifie que vous êtes en mesure d'exécuter les commandes MS.DOS, comme si vous étiez sous le contrôle de ce système d'exploitation.

Les fichiers programmes associés à ces commandes doivent cependant avoir été préalablement convertis, à l'aide de l'utilitaire LMS en fichiers programmes de format PROLOGUE de données, ou les fichiers programmes utilisateur, convertis eux aussi, sont, par contre, de type quelconque.

Syntaxe :

[unité.]MSDOS <rc>

Il apparaît alors sur l'écran la bannière du Décor, la correspondance entre les unités de MS.DOS et celles de PROLOGUE, et l'unité par défaut.

```
tm
DECOR MS.DOS Version 1.xx
Copyright (c) 1983 BULL MICRAL
Msdos/Prologue Units Equals :
*A: < = > FL0
*B: < = > FL1
```

A >

Le déroulement des opérations :

Dès l'apparition du signal d'appel A >, vous vous trouvez dans un environnement MS.DOS ; vous pouvez alors procéder à l'exécution, dans la syntaxe MS.DOS, des commandes suivantes : DIR, ERA, REN, TYPE, et de certains utilitaires MS.DOS qui auront été convertis.

Les caractères séparateurs peuvent être choisis indifféremment selon les règles propres à MS.DOS ou à PROLOGUE.

Exemple :

Les commandes :

```
A > WM, PROG01.ASM <rc>
A > WM, PROG01-ASM <rc>
A > WM PROG01.ASM <rc>
A > WM PROG01-ASM <rc>
```

ont des effets identiques.

6.2.3.2. La commande LMS-- : consultation du répertoire.

Objet de la commande :

La commande LMS-- vous permet de consulter sous PROLOGUE le répertoire d'un support de format MS.DOS.

Syntaxe 1 :

```
[unité.]LMS--,DIR,unité1 <rc>
```

où :

unité mnémonique de l'unité où se trouve le programme
 utilitaire LMS--. L'unité implicite système est prise
 par défaut.

unité1 mnémonique de l'unité où se trouve la disquette de
 format MS.DOS dont vous voulez consulter le
 répertoire.

Exemple :

-> LMS-- ,DIK <rc>

SYSTEME		40960	1-06-83	12:00p
SYSCONF	X	4480	1-06-83	12:00p
SYSERF	S	4096	1-06-83	12:00p
SYSERGB	S	4096	1-06-83	12:00p
SYSTOO	C	4096	1-06-83	12:00p
SYSCONF	S	4096	1-06-83	12:00p
/	X	19584	1-06-83	12:00p
CP	X	14336	1-06-83	12:00p
WSOVLYI	OVR	41216	1-06-83	12:00p
WSMSGs	OVR	31360	1-06-83	12:00p
WM	CMD	16384	1-06-83	12:00p
WS	CMD	21248	1-06-83	12:00p
WM	HLP	4096	1-06-83	12:00p
CPM	X	24704	1-06-83	12:00p
DECOR	BAK	16896	1-06-83	12:00p
DECOR		17408	1-06-83	12:00p
SPL01	X	8192	1-06-83	12:00p

Attention :

LMS-- est un utilitaire de PROLOGUE. Vous devez respecter la syntaxe des commandes et mnémoniques de PROLOGUE.

6.2.3.3. La commande LMS-- : Transfert des fichiers

Objet de la commande :

LMS-- permet également de transférer un fichier d'un support du format MS.DOS sur un support de format PROLOGUE.

Syntaxe 2 :

[unité.]LMS--,[unité1.]nomfich-DOS $\left[\begin{array}{l} \text{unité 2} \\ \text{ou} \\ \text{[unité2.]nomfich-PRL} \end{array} \right] <rc>$

où :

unité mnémonique de l'unité où se trouve le programme utilitaire LMS—. L'unité système est prise par défaut.

unité1 mnémonique PROLOGUE de l'unité où se trouve le support de format MS.DOS.

nomfich—DOS

nom du fichier MS.DOS à transférer sur son support PROLOGUE. Le séparateur entre le nom et le type peut être indifféremment celui de MS.DOS ou celui de PROLOGUE.

unité2 mnémonique PROLOGUE de l'unité où se trouve le support de format PROLOGUE qui reçoit le fichier transféré.

Si seule la mnémonique de l'unité2 est présente dans sa ligne de commande, le fichier sur le support PROLOGUE de destination aura le même nom que sur le support de format MS.DOS.

nomfich—PRL

nom attribué au fichier transféré sur le support PROLOGUE. Ce nom peut être différent de celui du fichier sur le support de format MS.DOS.

Remarque :

La taille maximale des fichiers pouvant être transférés est de 512 K octets.

Exemples :

```
-> LMS—,MBASIC-COM,MD0 <rc>  
-> LMS—,FL0,BASCOM-COM,MD0 <rc>  
-> LMS—,FL0.TOTO-ASM <rc>  
-> LMS—,FL1.TUTU-BAS,TATA-BAS <rc>
```

6.3. LE SEQUENTIEL INDEXE

6.3.1. Généralités

La méthode d'accès séquentielle indexée génère des fichiers séquentiels indexés qui sont composés, en général, de deux fichiers N.G.F. physiquement disjoints, mais logiquement liés :

- Un fichier des clés,
- Un fichier des données (facultatif).

La sauvegarde et la restauration d'un fichier séquentiel indexé se font au moyen de deux utilitaires qui sont employés uniquement pour ces types de fichiers. Pour les utiliser, il n'est pas nécessaire d'avoir chargé auparavant la méthode d'accès séquentielle indexée par SYSCONF (voir paragraphe 7.4.2. — Configuration des méthodes d'accès), car ces deux utilitaires n'y font pas d'accès.

Hormis, l'intérêt d'une sauvegarde proprement dite, faire la sauvegarde puis la restauration d'un fichier S.I. peut permettre de vérifier l'état du fichier et de le réorganiser. Notamment, disparaîtront les chaînes des vides, d'où gain de place sur l'unité et les clés seront réorganisées, d'où gain de temps de traitement.

6.3.2. Sauvegarde des fichiers séquentiels indexés : “SVSI”

Commande d'appel :

[unité.]SVSI,desfic1[,DEST=desfic2]<rc>

où :

- | | |
|---------|---|
| unité | Nom de l'unité où se trouve l'utilitaire SVSI.
Par défaut, il sera recherché sur l'unité implicite programme. |
| SVSI | Nom du fichier binaire correspondant, ayant comme suffixe “X”. |
| desfic1 | Désigne le fichier séquentiel indexé à sauvegarder.
Par défaut, c'est le fichier ayant un suffixe “I” qui sera recherché. S'il existe des données associées, c'est le fichier ayant un suffixe “D” qui lui sera associé. |
| desfic2 | Désigne le fichier séquentiel de sauvegarde (fichier destination). Son suffixe, s'il n'est pas spécifié ou si ce paramètre est omis, est “U”. Le nom du fichier, par défaut, est celui du fichier d'origine. |

Remarques :

- Si le fichier de destination n'existe pas, il sera automatiquement créé. S'il existe, il sera écrasé.
- Une sauvegarde peut se faire sur plusieurs volumes. Avant de la lancer, assurez-vous que tous les volumes dont vous aurez besoin sont préalablement formatés et qu'ils sont des volumes PROLOGUE. SVSI ne fait ni formatage, ni création de volume.
- Durant la sauvegarde, les fichiers sont ouverts en mode exclusif. Elle exige donc que ces fichiers ne soient pas utilisés par un autre poste.
- Après la sauvegarde, le fichier séquentiel indexé d'origine n'est pas détruit.

Exemple 1 :

-> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0
-> SVSI,FICSI,DEST=FL0<rc>

L'utilisateur souhaite sauvegarder le fichier séquentiel indexé "FICSI" se trouvant sur l'unité implicite fichier "MD0", qui se compose donc du fichier des clés. "FICSI-I" et éventuellement du fichier des données "FICSI-D". Le fichier de sauvegarde (destination) se trouvera sur l'unité "FL0" et portera le nom de "FICSI-U".

MESSAGES D'EXECUTION

SVSI n'affiche pas de message particulier, sauf pour gérer la sauvegarde multivolume. Dans tous les cas, une fois la sauvegarde du fichier séquentiel indexé terminée, l'utilitaire affiche un "-" indiquant que la sauvegarde s'est effectuée correctement.

Exemple :

-> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0
-> SVSI,FICSI -
-> -

L'utilisateur a lancé une sauvegarde du fichier séquentiel indexé "FICSI" se trouvant sur l'unité implicite fichier "MD0". Le fichier de destination "FICSI-U" a été créé sur cette même unité. La sauvegarde s'est effectuée sans incident, l'utilitaire ayant affiché le caractère "-".

* "Monter le volume suivant, Prêt (o/n) :

Ce message apparaît lorsque le fichier de sauvegarde ne peut pas tenir sur un seul volume. Après l'affichage de ce message, l'utilisateur doit monter un autre volume puis frapper la lettre "O" suivie de <rc>, s'il veut poursuivre sa sauvegarde. Dans le cas où il souhaite abandonner la sauvegarde de son fichier séquentiel indexé, il peut frapper "N" suivi de <rc>. Dans ce cas, il retournera à l'interpréteur de commandes et le fichier de sauvegarde NE SERA PAS DETRUIT.

Exemple 1 :

—> SVSI,FICSI,DEST=FL0 Monter le volume suivant, Prêt (o/n) : O —
—> —

La sauvegarde s'est effectuée sur deux volumes différents et s'est bien terminée puisque le "—" a été affiché.

Exemple 2 :

—> SVSI,FICSI,DEST=FL0.SFICSI Monter le volume, Prêt (o/n) : N
Erreur Fichier n° 46 Débordement Volume
—> —

Au moment de changer de volume, l'utilisateur a abandonné en frappant "N". La sauvegarde n'a pu être menée à son terme et le "—" n'a donc pas été affiché.

MESSAGES D'ERREUR

En plus des messages système dont vous trouverez la signification dans l'annexe B, SVSI émet ses propres messages d'erreur.

* "Fichier inexistant"

Ce message apparaît lorsque SVSI n'a pu trouver le fichier spécifié à sauvegarder.

Exemple :

—> SVSI,FICS Fichier inexistant
—> —

"FICS-I" n'existe pas sur l'unité implicite fichier.

* "Syntaxe incorrecte"

Ce message apparaît lorsque la syntaxe de la commande est inexacte.

Exemple :

—> SVSI,FICS* Syntaxe incorrecte

SVSI n'admet pas les caractères de substitution, car il ne gère pas la sauvegarde d'un groupe de fichiers séquentiels indexés.

* "Fichier ouvert"

Ce message apparaît lorsque le fichier à sauvegarder est déjà ouvert par un autre poste.

Exemple :

—> SVSI,FICSI Fichier ouvert

—> _

6.3.3. Restauration des fichiers séquentiels indexés : "RTSI"

Commande d'appel :

```
[unité.]RTSI,desfic1[,DEST=desfic2]<rc>
```

où :

- | | |
|---------|--|
| unité | Nom de l'unité où se trouve l'utilitaire RTSI.
Par défaut, il sera recherché sur l'unité implicite programme. |
| RTSI | Nom du fichier binaire correspondant, ayant comme suffixe "X". |
| desfic1 | Désigne le fichier de sauvegarde à restaurer.
Par défaut, le suffixe est "U". |
| desfic2 | Désigne le fichier séquentiel indexé résultant de la restauration.
Par défaut, le fichier des clés restauré aura le même nom que le fichier de sauvegarde, mais le suffixe sera "I". Quant au fichier des données, s'il existe, son suffixe sera "D". Si le nom de fichier restauré est spécifié sans suffixe, nous aurons respectivement les mêmes types. Si le suffixe est spécifié dans la commande, seul le fichier des clés aura ce suffixe. |

Remarques :

- L'utilitaire RTSI crée le fichier des clés et, s'il y a lieu, le fichier des données. S'ils existent déjà, ils sont écrasés.
- Si la sauvegarde a été faite sur plusieurs volumes, il conviendra de présenter les différents volumes dans l'ordre où ils ont été utilisés lors de la sauvegarde.
- Après la restauration, le fichier d'origine ne sera pas détruit.

Exemples :

—> RTSI,CLIENT

Un fichier CLIENT-I contiendra les clés et un fichier CLIENT-D contiendra les données. La restauration est faite à partir du fichier "CLIENT-U".

—> RTSI,CLIENT,DEST=FACTURE

Un fichier FACTURE=I contiendra les clés et un fichier FACTURE-D contiendra les données. La restauration est faite à partir du fichier "CLIENT-U".

—> RTSI,CLIENT-,DEST=FACTURE-Z

Un fichier FACTURE-Z contiendra les clés et un fichier FACTURE-D contiendra les données. La restauration est faite à partir du fichier "CLIENT-".

A NE PAS FAIRE :

Si les fichiers de sauvegarde FIC01 et FIC02 ont des données :

—> RTSI,FIC01,DEST=CLIENT-83

—> RTSI,FIC02,DEST=CLIENT-84

aura pour conséquence de créer un fichier CLIENT-D pour les données des deux fichiers. Mais on ne retrouve que les données du deuxième fichier, puisque celles appartenant au premier auront été écrasées.

MESSAGES D'EXECUTION

RTSI n'affiche pas de message particulier, sauf pour gérer la restauration multivolume. Dans tous les cas, une fois la restauration du fichier séquentiel indexé terminée, l'utilitaire affiche un "—" indiquant que la restauration s'est effectuée correctement.

Exemple :

—> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0
—> RTSI,FICSI —
—> _

L'utilisateur a lancé une restauration du fichier séquentiel indexé "FICSI-U" se trouvant sur l'unité implicite fichier "MD0".

Le fichier séquentiel indexé de destination composé de "FICSI-I" (fichier des clés) et "FICSI-D" (fichier des données) a été créé sur cette même unité. La restauration s'est effectuée sans incident, l'utilisateur ayant affiché le caractère "—".

* "Monter le volume suivant, Prêt (o/n) :"

Ce message apparaît lorsque la sauvegarde a été effectuée sur plusieurs volumes. Après l'affichage de ce message, l'utilisateur doit monter un autre volume puis frapper la lettre "O" suivie de <rc>, s'il veut poursuivre sa restauration. Dans le cas où il souhaite abandonner la restauration de son fichier séquentiel indexé, il peut frapper "N" suivi de <rc>. Dans ce cas, il retournera à l'interpréteur de commandes et le fichier séquentiel indexé, partiellement restauré, sera SUPPRIME.

Exemple 1 :

—> RTSI,FL0.FICSI Monter le volume suivant, Prêt (o/n) : O —
—> _

La restauration s'est effectuée à partir de deux volumes différents montés successivement sur l'unité "FL0" et s'est bien terminée puisque le "—" a été affiché.

Exemple 2 :

—> RTSI,FL0.FICSI,DEST=FICSI1 Monter le volume suivant, Prêt (o/n) : N
Erreur Fichier n° 48 Fin de Fichier
—> _

Au moment de changer de volume, l'utilisateur a abandonné en frappant "N". La restauration n'a pu être menée à son terme, l'utilisateur a signalé à l'utilisateur une fin de fichier anormale et le "—" n'a donc pas été affiché.

MESSAGES D'ERREUR

En plus des messages système dont vous trouverez la signification dans l'annexe B, RTSI émet ses propres messages d'erreur.

* "Fichier inexistant"

Ce message apparaît lorsque RTSI n'a pu trouver le fichier spécifié à restaurer.

Exemple :

—> RTSI,FICS Fichier inexistant
—>

"FICS-U" n'existe pas sur l'unité implicite fichier.

* "Syntaxe Incorrecte"

Ce message apparaît lorsque la syntaxe de la commande est inexacte.

Exemple :

—> RTSI,FIC* Syntaxe incorrecte

RTSI n'admet pas les caractères de substitution, car il ne gère pas la restauration d'un groupe de fichiers séquentiels indexés.

* "fichier ouvert"

Ce message apparaît lorsque le fichier-à sauvegarder est déjà ouvert par un autre poste.

Exemple :

—> RTSI,FICSI Fichier ouvert
—>

Le fichier "FICSI-U" est ouvert par un autre poste.

6.4. LE MULTI-CRITERES

6.4.1. Généralités

La méthode d'accès "Multi-critères" génère des fichiers N.G.F. physiquement disjoints, mais logiquement liés :

- fichiers des clés : F1-I,
- fichier des données : F1-D,
- fichier de description des liens, des rubriques... : F1-V,
- fichier des références : F1-R.

La sauvegarde et la restauration des fichiers multi-critères se font au moyen des utilitaires SVSI et RTSI.

6.4.2. Sauvegarde de fichiers Multi-critères

Commandes à exécuter :

- > SVSI,F1,DEST=F1-IU
- > SVSI,F1-V,DEST=F1-VU
- > SVSI,F1-R,DEST=F1-RU

Les messages d'exécution et d'erreurs sont ceux du séquentiel indexé.

6.4.3. Restauration de fichiers Multi-critères

Commandes à exécuter :

- > RTSI,F1-IU
- > RTSI,F1-VU,DEST=F1-V
- > RTSI,F1-RU,DEST=F1-R

Les messages d'exécution et d'erreurs sont ceux du séquentiel indexé.

6.5. LA BASE DE DONNEES

6.5.1. Généralités

La méthode d'accès "Base de données" génère des fichiers multi-critères et un dictionnaire, physiquement disjoints mais logiquement liés :

Base de données : BASE — B

Fichiers : F1 — I
F1 — D
F1 — V
F1 — R

F2 — I
F2 — D
F2 — V
F2 — R

Jointure : J1 — I
J1 — D
J1 — V
J1 — R

La sauvegarde et la restauration des bases de données se font au moyen des utilitaires SVSI et RTSI.

6.5.2. Sauvegarde d'une base de données

Commandes à exécuter :

—> SVSI,BASE-B,DEST=BASE-BU	}	Sauvegarde du dictionnaire
—> SVSI,F1,DEST=F1-IU		
—> SVSI,F1-V,DEST=F1-VU	}	Sauvegarde des fichiers
—> SVSI,F1-R,DEST=F1-RU		
—> SVSI,F2,DEST=F2-JU		
—> SVSI,F2-V,DEST=F2-VU		
—> SVSI,F2-R,DEST=F2-RU	}	Sauvegarde des jointures
—> SVSI,J1,DEST=J1-IU		
—> SVSI,J1-V,DEST=J1-VU		
—> SVSI,J1-R,DEST=J1-RU		

Les messages d'exécution et d'erreurs sont ceux du séquentiel indexé.

6.5.3. Restauration d'une base de données

Commandes à exécuter :

-> RTSI,BASE-BU,DEST=BASE-B	}	Restaure le dictionnaire
-> RTSI,F1-IU		
-> RTSI,F1-VU,DEST=F1-V		
-> RTSI,F1-RU,DEST=F1-R	}	Restaure les fichiers
-> RTSI,F2-IU		
-> RTSI,F2-VU,DEST=F2-V		
-> RTSI,F2-RU,DEST=F2-R	}	Restaure la jointure
-> RTSI,J1-IU		
-> RTSI,J1-VU,DEST=J1-V		
-> RTSI,J1-RU,DEST=J1-R		

Les messages d'exécution et d'erreurs sont ceux du séquentiel indexé.

6.6. MISE EN ŒUVRE DU B.A.L.

6.6.1. L'exécuteur : "EX"

Le B.A.L. (Basic Application language) est un langage de programmation orienté qui peut s'apparenter au BASIC mais qui diffère de celui-ci par la richesse d'instructions, la segmentation des programmes et la déclaration de toute variable, avant utilisation. Tous les programmes B.A.L. sont d'abord écrits dans un fichier source ayant comme suffixe "S", au moyen d'un éditeur de textes, puis sont traduits avec l'utilitaire "TR-X" pour donner un fichier dont le suffixe est "T". L'exécution de ce dernier se fait au moyen de l'utilitaire "EX-X".

L'utilitaire EX peut être installé de manière permanente en mémoire, lors de l'initialisation du système. Dans ce cas, son exécution se fait sans chargement préalable du fichier correspondant et il y a partage de l'utilitaire entre les différents postes utilisateurs. Cette installation se fait pas la déclaration suivante, dans le fichier source SYSCONF-S :

- EX < INTR

Pour plus de détails, reportez-vous au paragraphe 7.4.5. "configuration des utilitaires résidents " du chapitre 7.

Commande d'appel :

Première syntaxe (à partir de la version 4.8a) :

-> [[unité.]EX,]desfic[,paramètres[:[DB][,CLR]]<rc>

Deuxième syntaxe (pour les versions antérieures à la 4.8a) :

-> [[unité.]EX,]desfic[,DB]<rc>

où :

unité	Indique l'unité sur laquelle l'utilitaire EX est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire est recherché sur l'unité implicite programme de votre poste.
EX	Nom du fichier binaire correspondant, de suffixe "X". Il désigne l'exécuteur B.A.L. S'il est omis, l'interpréteur de commande le chargera à partir de la même unité que le fichier traduit à exécuter.
desfic	Descripteur du fichier à exécuter. son suffixe est "T" obligatoirement. Si EX n'est pas spécifié dans la commande, le suffixe ne devra pas l'être non plus.
paramètres	Chaînes de caractères séparées par des virgules, utilisées comme valeur d'initialisation par les programmes. Cette option n'existe qu'à partir de la version 5.1 de l'exécuteur.
DB	Signifie que le programme B.A.L. doit être exécuté sous le contrôle du moniteur de mise au point.
CLR	Spécifie que l'effacement de l'écran doit être forcé par l'exécuteur avant de donner le contrôle au programme. Cette option ne tient pas compte de la directive "NOCLR" éventuellement contenue dans le programme B.A.L. Cette option n'existe qu'à partir de la version 5.1 de l'exécuteur.

Remarques :

- L'option "DB" n'offre d'intérêt que si on possède le programme source traduit, avec les adresses d'implantation des différentes instructions.
- Les options disponibles à partir de la version 5.1 de l'exécuteur ne pourront être utilisées que si le programme B.A.L. a été traduit avec un traducteur dont la version est aussi 5.1.

Exemple 1 :

-> Prologue V2.3 Poste 1 Implicites : FL0,MD0,IM0
-> EX,PROGBAL <rc>

L'utilitaire "EX-X" sera chargé à partir de l'unité implicite programme "MD0" et le programme B.A.L. "PROGBAL-T" sera recherché sur l'unité implicite fichier "FL0".

Exemple 2 :

-> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0
-> PROGBAL<rc>

Le programme B.A.L. "PROGBAL-T" sera recherché sur l'unité implicite programme "MD0" par l'interpréteur qui chargera l'exécuteur à partir de la même unité.

MESSAGES D'EXECUTION ET MESSAGES D'ERREUR

L'exécuteur B.A.L. n'affiche aucun message lors de l'exécution d'un programme B.A.L. Par contre, il peut afficher des messages d'erreur. Ceux-ci sont spécifiques au B.A.L. et sont alors de la forme :

Erreur Système n° xxx (B.A.L.) à l'adresse yyyy (seg : z)

où :

xxx	Est le numéro de l'erreur B.A.L. (voir annexe B).
yyy	Est l'adresse hexadécimale de mise au point de l'instruction où s'est produite l'erreur.
z	Est le numéro de segment (du programme) en cours d'exécution.

On voit que ces messages d'erreurs sont donc intimement liés au programme. Pour plus de précision, il conviendra de se reporter au manuel "Langage B.A.L. Manuel de référence". Nous nous limiterons donc ici aux messages d'erreur envoyés par l'interpréteur de commandes refusant l'exécution d'un programme B.A.L.

* "xxxxxx : programme inexistant"

Ce message est affiché lorsque le programme B.A.L. "xxxxxx" spécifié dans la commande d'appel n'existe pas.

Exemple :

—> PRGBAL

PRGBAL : programme inexistant

—> _

Le programme “PRGBAL-T” n'existe pas sur l'unité implicite programme.

* “EX : programme inexistant”

Ce message est affiché lorsque le programme B.A.L. a été trouvé, mais pas l'exécuteur “EX-X”.

Exemple :

—> Prologue V2.3 Poste 1 Implicites : MD0,MD0,IM0

—> FL0.PROGBAL

EX : Programme inexistant

—> _

Le programme “PROGBAL-T” a bien été trouvé sur l'unité spécifiée “FL0”, mais l'exécuteur n'a pas été trouvé sur cette même unité. Rappelons que dans ce type de syntaxe, où l'exécuteur n'est pas cité dans la commande, “EX-X” doit se trouver OBLIGATOIREMENT sur le MEME support que le programme B.A.L. à exécuter.

* “Erreur de Commande n° 36 Structure Fichier Incorrecte”

Ce message est affiché lorsque le programme chargé ne peut être exécuté.

Exemple :

—> PROGBAL.T

Erreur de Commande n° 36 Structure fichier Incorrecte

—> _

Le fichier “PROGBAL-T” a été pris, non comme un programme B.A.L., mais comme un utilitaire. Dans une commande sans spécification de l'exécuteur, le nom du programme B.A.L. ne doit pas être suivi de son suffixe.

6.7. UTILITAIRES PROLOGUE MIS EN ŒUVRE PAR L'EXECUTEUR B.A.L.

6.7.1. Inhibition des granules ayant des secteurs invalides : "TOPGR-T".

Présentation :

L'utilitaire TOPGR-T, écrit en langage B.A.L., vous permet d'inhiber les granules d'un volume PROLOGUE, s'il y détecte un ou plusieurs secteurs invalides.

Commande d'appel :

—> EX,TOPGR

ou

—> TOPGR (si IP=IF)

Le système vous demande alors de rentrer le nom du support à contrôler (FL0, MD0, ...), lorsqu'il affiche la question :

nom de support :

Exécution :

Le programme analyse la table des granules, puis accède à tous les secteurs des granules libres.

Si l'un d'eux retourne une erreur de lecture, le programme essaie d'écrire et de relire.

Si l'erreur persiste, le message :

"ERREUR XX GRANULE INHIBE"

est affiché à l'écran et le granule en cours est inhibé.

A la fin du traitement, le nombre de granules inhibés est affiché pour information.

A tout moment, il est possible d'abandonner le traitement en appuyant sur la touche "ESC". Le message

"ABANDON (O/N) :"

est affiché à l'écran.

Répondez "O" pour abandonner, "N" pour continuer.

Remarque :

Il est recommandé d'utiliser ce programme après un "CP,CV" de manière à inhiber les granules avant d'y mémoriser des informations.

6.7.2. Programmation des touches fonctions : "TFP"

Présentation :

Sur le clavier de votre terminal, il existe certaines touches libellées "Fn", où "n" est une valeur numérique : ce sont les touches fonctions. Leur nombre est variable suivant les types de claviers, mais toutes sont à la disposition de l'utilisateur seules ou associées à une autre touche, afin qu'il puisse les programmer, c'est-à-dire leur donner la valeur qu'il souhaite. Contrairement aux autres touches qui ne représentent qu'un caractère ASCII (dont la valeur hexadécimale est donc comprise entre 00 et FF inclus), les touches fonctions peuvent représenter un caractère ou une chaîne de caractères. Cette dernière peut être formée de n'importe quelle suite de caractères ASCII. La longueur maximale globale des chaînes de caractères affectées aux touches fonction dépend du micro-ordinateur utilisé. La programmation des touches fonctions est du plus grand intérêt lorsqu'on utilise, avec des fréquences élevées, les mêmes chaînes de caractères. Frapper une seule touche, au lieu d'une suite de caractères, limite les erreurs de frappe et augmente la vitesse de saisie.

Pour programmer ces touches fonctions PROLOGUE vous propose le programme B.A.L. "TFP".

-> [[unité1.] EX.] [unité2.] TFP <rc>

où

unité1	Indique l'unité sur laquelle l'utilitaire EX est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire est recherché sur l'unité implicite programmé de votre poste.
EX	Nom du fichier binaire correspondant, de suffixe "X". Il désigne l'exécuteur B.A.L. S'il est omis, l'interpréteur de commande le chargera à partir de la même unité que le fichier à exécuter.
unité2	Indique l'unité sur laquelle le programme TFP est recherché afin d'être exécuté. Si l'unité est omise et EX non spécifié dans la commande, le programme est recherché sur l'unité implicite programme de votre poste. Si l'unité est omise et EX spécifié, le programme est recherché sur l'unité implicite fichier de votre poste.
TFP	Nom du fichier à exécuter. De suffixe "T".

Déroulement des opérations :

Une fois la commande prise en compte par le système, TFP efface l'écran puis affiche le premier écran suivant :

Programmation des touches fonctions Version 1.2a
Copyright © 1985 BULL Micral

RAZ de toutes les touches tapez 0

Programmation normale tapez N

Chargement des fichiers TFP tapez C

Option choisie

Donc, après avoir frappé "O" ou "N" apparaît un second écran qui va vous permettre de programmer chaque touche fonction, une à une :

PROGRAMMATION DES TOUCHES FONCTIONS

touche 1 :
touche 2 :
touche 3 :
touche 4 :
touche 5 :
touche 6 :
touche 7 :
touche 8 :
touche 9 :
touche 10 :
touche 11 :

Numéro de la touche fonction à programmer : —

(flèche basse = page suivante - flèche haute = page précédente)
(LINE FEED ou ^A permet de terminer la commande sans retour-
chariot)

Après l'affichage du tableau précédent, vous avez le choix entre les réponses suivantes :

- * Frapper ESC pour revenir à l'interpréteur de commandes.
- * Frapper la flèche basse : TFP affiche le tableau suivant, ainsi que les dix suivantes, si elles existent.
- * Frapper la flèche haute : TFP affiche alors le même tableau, mais avec le contenu de la première touche fonction, ainsi que les dix précédentes, si elles existent.
- * Frapper un numéro de touche suivi de `<rc>` : le curseur se positionne alors sur le premier caractère du contenu de la touche fonction, supprimant la visualisation de tous ceux qui, éventuellement, se trouveraient à sa droite. A partir de ce moment-là, vous pouvez programmer la touche fonction que vous avez choisie.

Programmation d'une touche fonction

Une fois que votre curseur est derrière le message "touche n :", où "n" représente le numéro de la touche fonction choisie, il vous suffit de rentrer votre chaîne de caractères suivie :

- d'un `<rc>`, si l'utilisateur souhaite que celui-ci soit inclus comme dernier caractère dans la programmation de la touche fonction. Un caractère "." est alors affiché derrière la chaîne de caractères.
- d'un CTRL-A, si l'utilisateur ne souhaite pas qu'un `<rc>` soit inclus comme dernier caractère dans la programmation de la touche fonction. Aucun caractère n'est alors affiché derrière la chaîne de caractères.

La frappe de `<rc>` ou CTRL-A provoque la modification immédiate de la touche fonction, puis le curseur vient se positionner derrière le message "Numéro de la touche fonction à programmer :", en attente de désignation d'une nouvelle touche fonction.

Lors de l'insertion des caractères de programmation de la touche fonction, il est possible d'insérer des caractères sous forme hexadécimale. Il suffit alors de frapper ESC. Le curseur viendra alors se positionner derrière le message "Code hexadécimal (01 à FE) :". Vous pourrez entrer le code choisi (attention, les caractères hexadécimaux 00, 03 et FF sont interdits). Une fois frappé, votre caractère sera affiché, s'il correspond à un caractère visualisable, ou sera remplacé par un ".", s'il ne l'est pas.

Vous pouvez ainsi, toujours lors de l'insertion de caractères, utiliser les flèches gauches et droites pour vous positionner sur le caractère à modifier. Avec la flèche gauche, le curseur se déplace d'un caractère vers la gauche, effaçant l'affichage du caractère se trouvant nouvellement à sa droite. Avec la flèche droite, le curseur se déplace d'un caractère à droite, si celui-ci existe et le faisant alors apparaître.

Remarques :

- Lorsque l'on veut supprimer la programmation d'une touche fonction, frapper en début un CTRL-A suffit.
- Lorsque le curseur se trouve sur un caractère non visualisable représenté par un ".", il affiche son équivalent hexadécimal à droite.
- 50 touches sont programmables.

Exemple de programmation des touches fonctions :

PROGRAMMATION DES TOUCHES FONCTIONS

touche 1 : MD0./,MD0
 touche 2 : MD0./,MD0,GR.
 touche 3 : MD9./,FL0.
 touche 4 : MD0./,FL0,GR.
 touche 5 : MD0.VPMI,IPRG=MD0,IFIC=MD0.
 touche 6 : MD0.VPMI,IPRG=FL0,IFIC=FL0.
 touche 7 : MD0.CP.DF,MD0.
 touche 8 : ,FL0.
 touche 9 :
 touche 10 :
 touche 11 :

Numéro de la touche fonction à programmer : —

(flèche basse = page suivante - flèche haute = page précédente)
 (LINE FEED ou ^A permet de terminer la commande sans retour-chariot)

Les touches 1, 2, 3 et 4 ont été programmées pour visualiser le catalogue des volumes situés en MD0 ou FL0, en ne frappant qu'une seule touche, la touche fonction. On peut remarquer la double signification des caractères ".".

Par exemple, pour la touche numéro 1, le premier est réellement un point. Par contre, le deuxième est un <rc>. Si notre curseur était dessus, on aurait l'affichage suivant :

touche 1 : MD0./,MD0. 0D

Les touches 5 et 6 ont été programmées pour modifier les unités implicites de poste. L'avantage est qu'en une seule touche on a exécuté VPMI, alors que lorsque la commande est frappée, celle-ci est souvent refusée parce que la syntaxe est incorrecte, l'unité ou VPMI n'étant pas spécifiée,...

Les touches 7 et 8 sont intimement liées. Elles permettent de recopier un fichier de l'unité "MD0" sur l'unité "FL0", en frappant seulement :

touche fonction 7	nom du fichier	touche fonction 8
-------------------	----------------	-------------------

On peut remarquer, entre CP et DF (touche fonction 7), un ".". Il correspond au <rc> qui normalement est frappé après la commande pour passer en mode répétitif (voir paragraphe 4.2.2. du chapitre 4.). Manipulation des fichiers et des volumes : "CP" du Guide Utilisateur PROLOGUE. De même, deux "." représentant deux <rc> sont utilisés dans la touche fonction 8 (derrière FL0) pour sortir de ce mode répétitif et ainsi revenir à l'interpréteur de commandes.

Messages d'erreur

Sur toute erreur de manipulation de la part de l'utilisateur, TFP émet un message sonore et le curseur reste immobile.

Toutes les valeurs des touches fonctions sont stockées dans un espace mémoire de longueur fixe. Lorsqu'au cours de la programmation d'une touche fonction la limite de cet espace mémoire est dépassée, TFP affiche le message "Débordement du tampon des touches fonctions", réinitialise le contenu de cette touche fonction, puis attend la frappe d'un <rc> de la part de l'utilisateur.

Celui-ci devra réduire alors la longueur d'une autre touche fonction, s'il veut pouvoir programmer sans restriction la touche fonction initiale.

Relation entre les numéros des touches dans TFP et le clavier.

	MICRAL 30 et compatibles									
Numéro TFP	1	2	3	4	5	6	7	8	9	10
Touche Clavier	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Numéro Touche KBF(), prfo	/80 128	/81 129	/82 130	/83 131	/84 132	/85 133	/86 134	/87 135	/88 136	/89 137
Numéro TFP	11	12	13	14	15	16	17	18	19	20
Touches Clavier	Shift F1	Shift F2	Shift F3	Shift F4	Shift F5	Shift F6	Shift F7	Shift F8	Shift F9	Shift F10
Numéro Touche KBF(), prfo	/8A 138	/8B 139	/8C 140	/8D 141	/8E 142	/8F 143	/90 144	/91 145	/92 146	/93 147
Numéro TFP	21	22	23	24	25	26	27	28	29	30
Touches Clavier	Ctrl F1	Ctrl F2	Ctrl F3	Ctrl F4	Ctrl F5	Ctrl F6	Ctrl F7	Ctrl F8	Ctrl F9	Ctrl F10
Numéro Touche KBF(), prfo	/94 148	/95 149	/96 150	/97 151	/98 152	/99 153	/9A 154	/9B 155	/9C 156	/9D 157
Numéro TFP	31	32	33	34	35	36	37	38	39	40
Touches Clavier	Alt F1	Alt F2	Alt F3	Alt F4	Alt F5	Alt F6	Alt F7	Alt F8	Alt F9	Alt F10
Numéro Touche KBF(), prfo	/9E 158	/9F 159	/A0 160	/A1 161	/A2 162	/A3 163	/A4 164	/A5 165	/A6 166	/A7 167
Numéro TFP	41	42	43	44	45	46	47	48	49	50
Touche[s] Clavier	PgUp	End	PgDn	Ins	Del	Ctrl PgUp	Ctrl End	Ctrl PgDn	Ctrl ←	Ctrl →
Numéro Touche KBF(), prfo	/A8 168	/A9 169	/AA 170	/AB 171	/AC 172	/AD 173	/AE 174	/AF 175	/B0 176	/B1 177

6.8. LE GRAPHIQUE

6.8.1. Généralités

Le graphique sous PROLOGUE vous offre des fonctions graphiques portables entre les différents modèles de la gamme BULL MICRAL. Il vous permet de représenter des valeurs sous forme graphique (voir schéma 8-A).

Ces fonctions sont proches de la norme G.K.S. (Graphical Kernel System) qui répond au besoin de standardisation dans l'utilisation du graphique.

Le graphique est une extension du système d'exploitation PROLOGUE et c'est à ce niveau que vous disposez de fonctionnalités graphiques générales et indépendantes des particularités des différents matériels.

Elles sont accessibles par le langage B.A.L. dans lequel des verbes graphiques ont été intégrés.

D'autres langages peuvent également y accéder par des procédures de type CALL, avec une interface écrit en assembleur.

Toutes ces fonctions sont décrites en détail dans les manuels de langages.

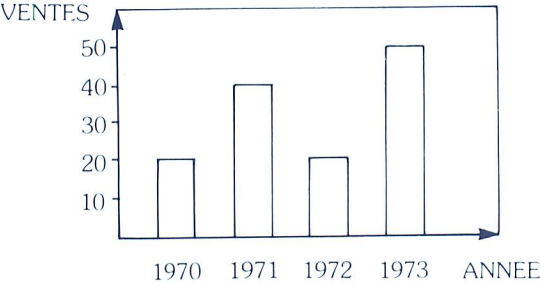
REPRESENTATION GRAPHIQUE DE DONNEES A PARTIR D'UN TABLEAU DE VALEURS

1) DIAGRAMME LINEAIRE



Schéma 8-A.

2) HISTOGRAMME



3) DIAGRAMME CIRCULAIRE

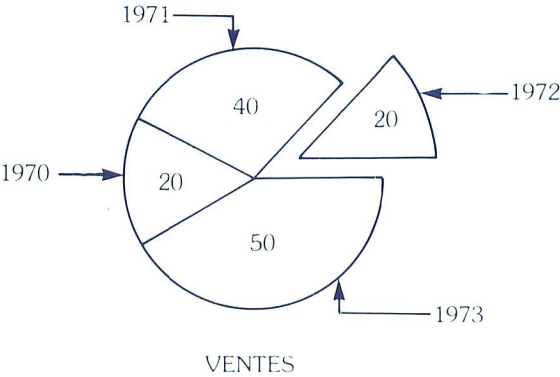


Schéma 8-A.

6.8.2. Description des dessins

6.8.2.1. Espace de travail/Espace de visualisation

6.8.2.1.1. Espace de travail

Les applications graphiques traitées sont des applications en deux dimensions.

Vous disposez d'un espace virtuel de travail ou espace utilisateur, défini dans un repère orthonormé dont les coordonnées sont des entiers signés variant de -32768 à $+32767$ en X et en Y. Le système de coordonnées dans lequel les dessins sont décrits est appelé système de coordonnées utilisateur.

Dans cet espace de travail, vous disposez d'une fenêtre qui en délimite une partie rectangulaire dont les côtés sont parallèles aux axes du système de coordonnées utilisateur. La taille de la fenêtre est inférieure ou égale à tout l'espace et ses dimensions sont précisées dans le système de coordonnées.

Seule la partie du dessin qui se trouve à l'intérieur de la fenêtre est affichée. Le système effectue un découpage de tous les traits par rapport à cette fenêtre (voir schéma 8-B).

6.8.2.1.2. Espace de visualisation

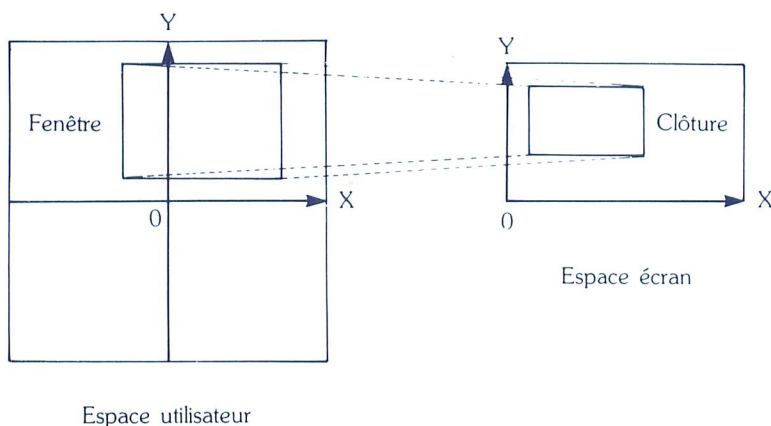
Dans l'espace de visualisation (le plus souvent l'écran), vous disposez d'une clôture qui en délimite une partie rectangulaire dont les côtés sont parallèles aux bords de cet espace. La taille et ses dimensions sont précisées dans le système de coordonnées de ce dernier.

Les coordonnées de l'espace de visualisation, ou coordonnées réelles, sont des entiers positifs de valeur inférieure ou égale aux dimensions de celui-ci. Par convention, le point origine de coordonnées ($x = 0, Y = 0$) est situé en bas à gauche de cet espace.

Tout au long de ce chapitre nous emploierons les termes d'espace écran pour l'espace de visualisation et de coordonnées écran pour les coordonnées réelles. Mais l'espace de visualisation peut désigner au autre support que l'écran (table traçante par exemple).

L'affichage consiste à projeter le contenu de la fenêtre dans la clôture.

Vous pouvez décrire vos dessins dans un espace virtuel de travail de façon indépendante de l'espace de visualisation (voir schéma 8-B).



TRANSFORMATION FENETRE-CLOTURE

Schéma 8-B

6.8.2.2. Fenêtre/clôture

En changeant indépendamment les dimensions de la fenêtre et de la clôture, vous pouvez obtenir des visualisations différentes de tout ou partie d'un même objet, par exemple des agrandissements et des réductions (différentes clôtures pour une même fenêtre) ou des effets de loupe (différentes fenêtres pour une même clôture).

Il est à noter que l'espace de travail est virtuel et qu'il n'existe pas de mémorisation des dessins affichés dans le système. Par conséquent, le changement de fenêtre ou de clôture n'entraîne aucune modification de l'image affichée. Il n'a pas d'effet immédiat sur la visualisation lors de la définition de fenêtre ou de clôture et il faut réexécuter les ordres de tracé qui composent les différents dessins pour que l'affichage en soit modifié (voir schéma 8-C).

DECOUPAGE D'UN DESSIN PAR RAPPORT A UNE FENETRE
ET PROJECTION DE CELLE-CI DANS DEUX CLOTURES DIFFERENTES

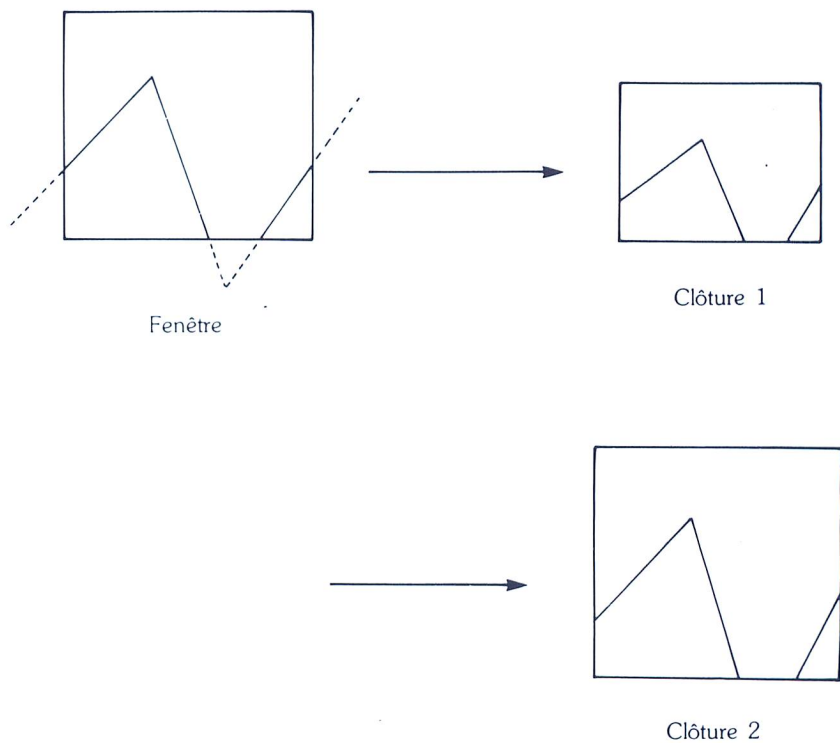
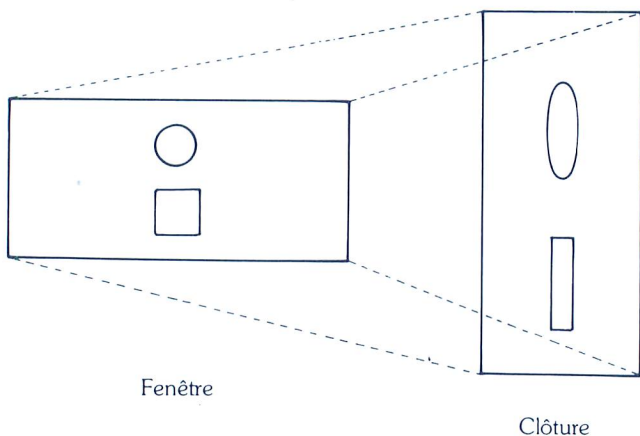


Schéma 8-C



DEFORMATION PARTIELLE LORS DE LA PROJECTION
DE LA FENETRE DANS LA CLOTURE
CE QUI ENTRAINE UNE INCOHERENCE PAR RAPPORT AU DESSIN INITIAL
DANS LA REPRESENTATION A L'ECRAN

Schéma 8-D

C'est pourquoi la convention suivante est adoptée :
La fenêtre est **toujours projetée sans déformation** dans la clôture.

Le dessin visualisé à l'écran a ainsi les mêmes proportions que celui décrit dans l'espace utilisateur, quelles que soient la fenêtre et la clôture.

Pour une fenêtre et une clôture données, le système graphique doit donc calculer les bornes de la clôture réelle dans laquelle la fenêtre est effectivement projetée.

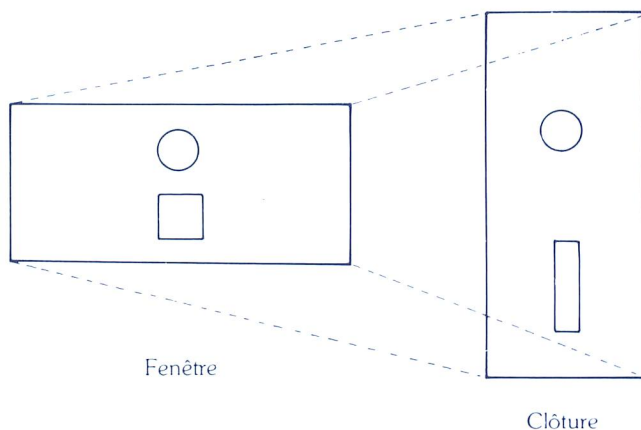
La clôture réelle est, non pas la clôture demandée dans le programme d'application, mais la plus grande clôture homothétique de la fenêtre, inscriptible dans la clôture d'origine et centrée par rapport à celle-ci (voir schéma 8-F).

6.8.2.3. Clôture réelle

La projection de la fenêtre dans la clôture peut entraîner des phénomènes de dilation. Par exemple, si une fenêtre étroite et haute est projetée dans une clôture large et basse, tous les dessins décrits dans la fenêtre vont subir une déformation lors de leur projection dans la clôture. De même qu'un carré dessiné dans l'espace utilisateur se transforme alors en rectangle à l'écran, un cercle devient une ellipse (voir schéma 8-D).

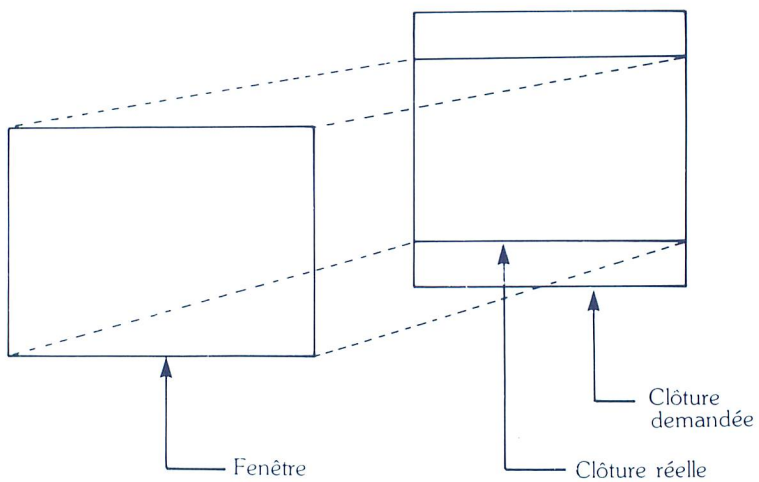
Or, pour des raisons d'efficacité et de temps de réponse, le système graphique laisse, dans la mesure du possible, les différents tracés tels que le cercle, à la charge de l'environnement.

On risque d'avoir, dans certains cas, une incohérence dans la représentation des dessins à l'écran. Notamment, la déformation que subissent les segments de droite par rapport aux dessins initiaux ne s'applique pas aux cercles (voir schéma 8-E).



DEFORMATION CORRECTE LORS DE LA PROJECTION
DE LA FENETRE DANS LA CLOTURE

Schéma 8-E



CALCUL DE LA CLOTURE REELLE

Schéma 8-F

6.8.3. Fonctionnalités du graphique

Toutes les fonctions graphiques sont associées à un numéro logique. Celui-ci permet d'identifier sans ambiguïté un ensemble de paramètres et de s'y référer.

Vous disposez actuellement de 32 numéros logiques.

6.8.3.1. Assignment d'un numéro logique

Avant tout ordre graphique, vous devez effectuer une assignation d'un numéro logique à une ressource périphérique (écran, table traçante,...). Par l'intermédiaire de ce numéro logique, les différents ordres de tracé sont associés au support périphérique sur lequel ils doivent être exécutés.

Une option d'exclusivité peut être précisée lors de cette assignation. Cette option se rapporte à la clôture associée au numéro logique et permet de réserver une partie de l'espace de visualisation à une application.

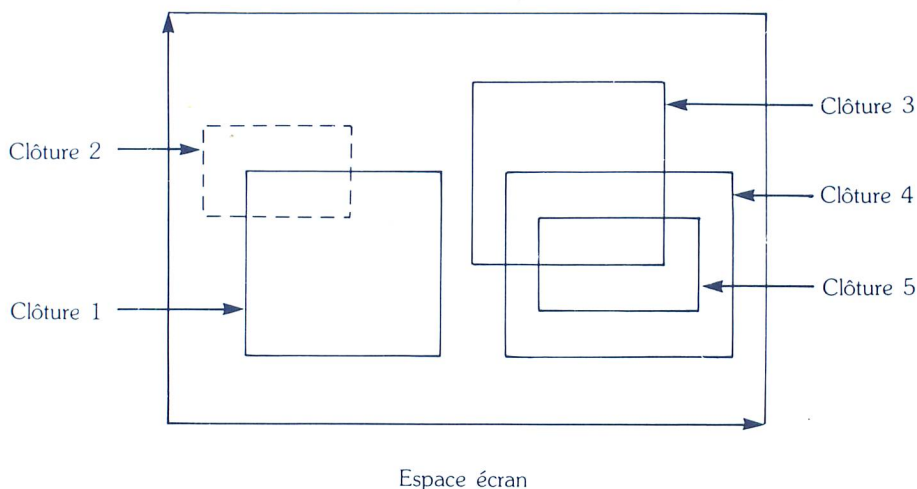
Deux clôtures non exclusives peuvent se chevaucher alors qu'une clôture exclusive ne peut ni recouvrir ni être recouverte par aucune autre.

Le système contrôle que la clôture définie exclusive ne recouvre partiellement ou totalement aucune clôture existante avec un numéro logique différent et par la suite qu'aucune nouvelle clôture ne recouvre celle-ci.

Si vous voulez positionner une clôture, exclusive ou non, qui recouvre une clôture exclusive existante sur le même support, le système graphique la rejette. Il en est de même pour une clôture exclusive qui recouvre une quelconque clôture existante.

Tout ordre de tracé avec un numéro logique associé à une telle clôture est rejeté en erreur par le système graphique tant que cette clôture n'est pas correctement définie.

Le caractère exclusif attribué à la clôture associée à un numéro logique ne prend fin qu'à la libération de ce dernier. Toutes les clôtures redéfinies entre temps avec ce numéro logique sont exclusives (voir schéma 8-G).



CLOTURES EXCLUSIVES ET PARTAGEABLES

LA CLOTURE 1 ETANT EXCLUSIVE, LA CLOTURE 2
EST PAR CONSEQUENT INCORRECTE
LES CLOTURES 3, 4 ET 5 SONT PARTAGEABLES

Schéma 8-G

6.8.3.2. Libération d'un numéro logique

Cette fonction permet de libérer un numéro logique précédemment assigné à une ressource périphérique.

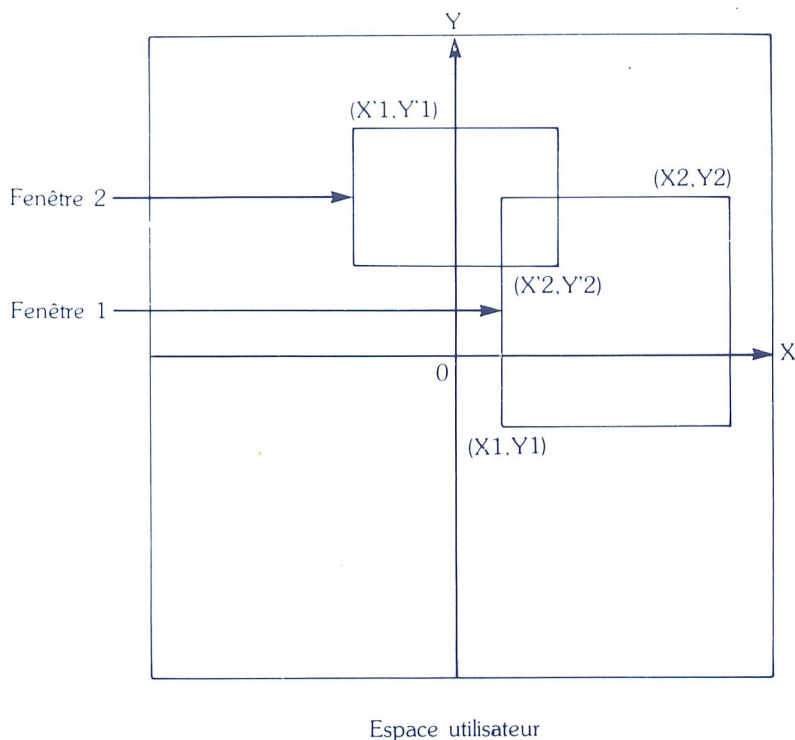
6.8.3.3. Définition de fenêtre et de clôture

Vous pouvez définir indépendamment une fenêtre dans l'espace utilisateur et une clôture dans l'espace de visualisation.

La fenêtre ou la clôture peuvent être définies, soit de façon explicite, soit de façon implicite.

Dans une définition explicite, vous précisez deux points diagonaux qui définissent la partie rectangulaire de l'espace de visualisation pour la clôture. Les deux points sont donnés dans un ordre quelconque et ce indifféremment, soit les coins en haut à gauche et en bas à droite. Les coordonnées des points sont données dans l'ordre X, Y.

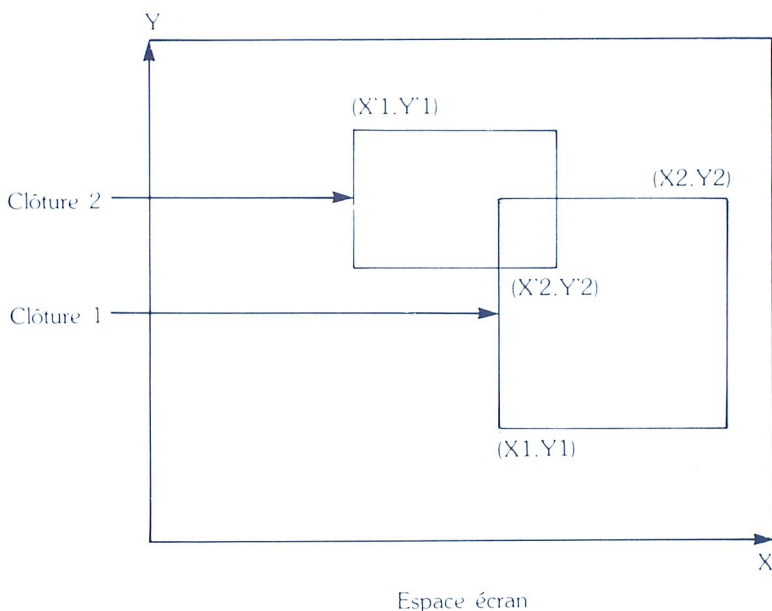
Les points définissant la fenêtre sont en coordonnées utilisateur, c'est-à-dire des entiers signés variant de -32768 à $+32767$ en X et en Y (voir schéma 8-H).



DEFINITION EXPLICITE DE FENETRES EN PRECISANT A CHAQUE FOIS DEUX POINTS DIAGONAUX

Schéma 8-H

Les points définissant la clôture sont en coordonnées réelles, c'est-à-dire des entiers positifs de valeur inférieure ou égale aux dimensions de l'espace de visualisation (voir schéma 8-I).



DEFINITION EXPLICITE DE CLOTURES, EN PRECISANT A CHAQUE FOIS DEUX POINTS DIAGONAUX

Schéma 8-1

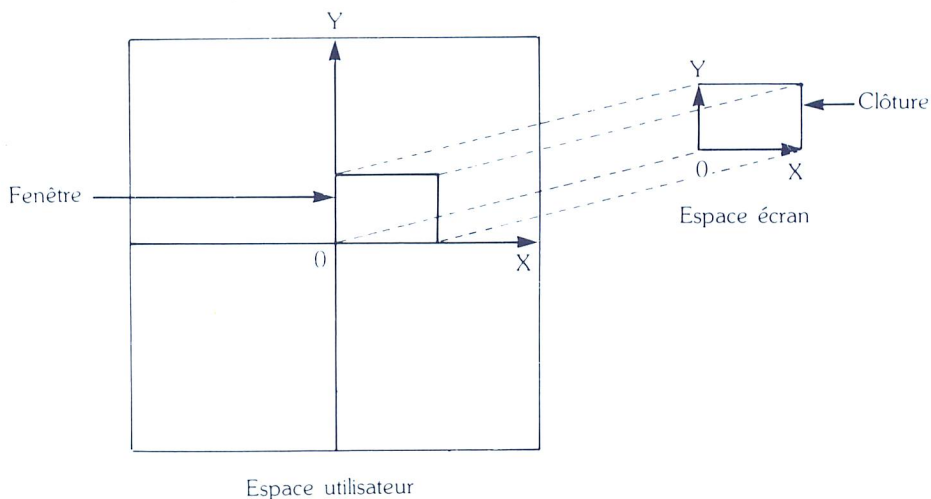
Dans une définition implicite, vous ne précisez aucun point et le système graphique prend pour taille de fenêtre ou de clôture, les dimensions de l'espace de visualisation (taille de l'écran). Le coin en bas à gauche d'une fenêtre définie de façon implicite est situé au point de coordonnées $(X=0, Y=0)$ de l'espace utilisateur.

Les définitions de fenêtre et de clôture sont envoyées avec un numéro logique.

Par l'intermédiaire de ce numéro logique, une fenêtre et une clôture sont associées. Ceci permet d'effectuer, à chaque ordre de tracé, la transformation en coordonnées réelles des points donnés en coordonnées utilisateurs.

Lors de l'assignation d'un numéro logique à une ressource périphérique, le système graphique prend implicitement pour valeur de fenêtre et de clôture associées à ce numéro, les dimensions de l'espace de visualisation (taille de l'écran). Le point de coordonnées ($X=0$, $Y=0$) de l'espace de visualisation (coin en bas à gauche de l'écran) correspondant au point de coordonnées ($X=0$, $Y=0$) de l'espace utilisateur.

Dans ce cas, comme dans le cas où de la fenêtre et la clôture ont toutes deux été définies de façon implicite par l'utilisateur, aucune transformation n'est effectuée sur les coordonnées des points donnés dans les ordres de tracé. Cela revient à décrire les dessins directement dans l'espace écran (voir schéma 8-J).



VALEUR IMPLICITES DE FENETRE ET DE CLOTURE (DIMENSIONS DE L'ESPACE DE VISUALISATION)

Schéma 8-J)

A chaque définition de fenêtre ou de clôture, si les dimensions de la clôture ne sont pas proportionnelles à celles de la fenêtre qui lui est associée, le système graphique modifie la clôture. Celui-ci calcule les bornes de la plus grande clôture inscrite dans celle demandée ou existante, de façon à ce que la projection de la fenêtre dans la clôture soit sans déformation.

Une erreur de modification de clôture vous est alors retournée.

Pour une définition de clôture, si cette dernière recouvre de façon incorrecte une clôture existante sur le même support, une erreur vous est retournée. Tout ordre de tracé associé à la nouvelle clôture est rejeté en erreur par le système graphique tant que celle-ci n'est pas correctement définie par l'utilisateur.

6.8.3.4. Ordres de tracé

L'affichage des dessins s'effectue au fur et à mesure des ordres de tracés.

Un ordre de tracé est envoyé avec un numéro logique et est ainsi associé à plusieurs paramètres :

- Le support périphérique (écran, table traçante,...) sur lequel le tracé est exécuté.
- Une fenêtre dans l'espace utilisateur et une clôture dans l'espace de visualisation, ce qui, permet d'effectuer la transformation en coordonnées réelles (coordonnées écran) des points donnés en coordonnées utilisateur dans l'ordre de tracé. De plus, un découpage du tracé est effectué par rapport à la fenêtre (clipping).
- Les attributs avec lesquels le tracé est exécuté. Ces attributs sont sélectionnés par une fonction indépendante.

Vous pouvez avoir trois attributs différents par tracé :

- un attribut de couleur
- un attribut de type
- un attribut de taille

Tous les points précisés dans les ordres de tracé sont en coordonnées utilisateur, c'est-à-dire des entiers signés variant de $-32768 + 32767$ en X et en Y. Les coordonnées des points sont données dans l'ordre X, Y.

6.8.3.4.1. Tracé de symboles

Cette fonction permet de tracer un même symbole à différents points donnés.

L'ordre est envoyé avec un tableau contenant les coordonnées utilisateur des différents points où le symbole doit être tracé.

Seuls les points intérieurs à la fenêtre associée sont tracés dans la clôture correspondante.

Par l'intermédiaire du numéro logique, trois attributs de tracé sont appliqués à l'affichage. Ce sont :

- la couleur du symbole
- le type du symbole sélectionné (point, croix,...)
- la taille du symbole

On peut envoyer jusqu'à 16384 points différents.

6.8.3.4.2. Tracé de lignes

Cette fonction permet de tracer des lignes entre différents points donnés.

L'ordre est envoyé avec un tableau contenant les coordonnées utilisateur des différents points à joindre par des lignes. Un découpage de ces lignes est effectué par rapport à la fenêtre associée, de façon à ne tracer, dans la clôture correspondante, que les segments de droite inférieurs à la fenêtre.

Par l'intermédiaire du numéro logique, trois attributs de tracé sont appliqués à l'affichage. Ce sont :

- la couleur de ligne
- le type de ligne sélectionné (continu, pointillé,...)
- l'épaisseur de ligne

On peut envoyer jusqu'à 16384 points différents et donc tracer jusqu'à 16383 segments de lignes brisées.

6.8.3.5. Attributs de tracé

Les ordres de tracé sont exécutés avec des attributs propres à chaque tracé.

Une valeur d'attribut de tracé se caractérise par un nombre. On peut avoir jusqu'à 256 valeurs différentes par attribut.

Les attributs de tracé sont sélectionnés par une fonction indépendante.

Lors de l'assignation d'un numéro logique à une ressource périphérique, le système graphique prend des valeurs implicites pour les différents attributs de tracé.

6.8.3.5.1. Attribut de couleur

Tous les ordres de tracé ont chacun un attribut de couleur associé. Le codage des 16 premières couleurs est établi de la façon suivante :

Valeur de l'attribut	Couleur correspondante
0	- noir
1	- rouge
2	- vert
3	- marron
4	- bleu
5	- magenta
6	- cyan
7	- gris clair
8	- gris foncé
9	- rouge clair
10	- vert clair
11	- jaune
12	- bleu clair
13	- magenta clair
14	- cyan clair
15	- blanc

Les valeurs d'attribut de couleur, au-delà de 15, seront définies ultérieurement.

Pour chaque ordre de tracé, si l'attribut de couleur n'a pas été sélectionné par l'utilisateur, le système graphique prend par défaut la couleur BLANC (valeur d'attribut = 15).

6.8.3.5.2. Attribut de type

6.8.3.5.2.1. Type de symbole

Cet attribut permet de sélectionner le type de symbole à tracer. Le codage des 15 premiers types de symbole est établi par le système graphique. Actuellement, seuls les 6 premiers types sont définis. Les types 7 à 15 sont réservés et seront définis ultérieurement.

Valeur de l'attribut	type de symbole correspondant
1	. (point)
2	+ (plus)
3	X (croix)
4	○ (rond)
5	□ (carré)
6	◇ (losange)

Les valeurs d'attribut de type de symbole, au-delà de 15, sont spécifiques à chaque environnement.

Si l'attribut de type de symbole n' a pas été sélectionné par l'utilisateur, le système graphique prend par défaut le type POINT (valeur d'attribut = 1).

6.8.3.5.2.2. Type de ligne

Cet attribut permet de sélectionner le type de ligne à tracer. Le codage des 15 premiers types de ligne est établi par le système graphique. Actuellement, seuls les 4 premiers types sont définis. Les types 5 à 15 sont réservés et seront définis ultérieurement.

Valeur de l'attribut	type de ligne correspondant
1	_____ (continu)
2	 (pointillé)
3	----- (tireté)
4	-. -. -. -. -. (tireté-pointillé)

Les valeurs d'attribut de ligne, au-delà de 15 sont spécifiques à chaque environnement.

Si l'attribut de type de ligne n'a pas été sélectionné par l'utilisateur, le système graphique prend par défaut le type CONTINU (valeur d'attribut = 1).

6.8.3.5.3. Attribut de taille

6.8.3.5.3.1. Taille de symbole

Cet attribut permet de sélectionner la taille des symboles à tracer.

Le codage des tailles de symboles est le suivant :

Valeur de l'attribut	taille de symbole correspondante
1	- simple taille
2	- double taille
3	- triple taille
.	.
.	.

La taille croît avec la valeur de l'attribut et peut varier de 1 à 255, suivant les possibilités de l'environnement.

Pour le symbole POINT, la simple taille (valeur d'attribut = 1) correspond à la taille du plus petit élément écran possible adressable (pixel).

Si l'attribut de taille de symbole n'a pas été sélectionné par l'utilisateur, le système graphique prend par défaut la simple taille (valeur d'attribut = 1).

6.8.3.5.3.2. Epaisseur de ligne

Cet attribut permet de sélectionner l'épaisseur de trait pour les lignes à tracer.

Le codage des épaisseurs de ligne est le suivant :

Valeur de l'attribut	Epaisseur de ligne correspondante
1	- simple épaisseur
2	- double épaisseur
3	- triple épaisseur

L'épaisseur croît avec la valeur de l'attribut et peut varier de 1 à 255, suivant les possibilités de l'environnement.

Si l'attribut d'épaisseur de ligne n'a pas été sélectionné par l'utilisateur, le système graphique prend par défaut la simple épaisseur (valeur d'attribut = 1).

6.8.3.6. Effacement de clôture

Cette fonction permet d'effacer la clôture associée au numéro logique. On peut préciser la couleur de fond avec laquelle on veut effacer la clôture écran. Si la couleur n'est pas précisée, le système graphique prend par défaut la couleur NOIR.

Il s'agit d'un effacement partiel ou total de l'écran, suivant la taille de la clôture associée au numéro logique.

6.8.3.7. Lecture des paramètres associés à un numéro logique

Cette fonction est envoyée avec un tableau que le système graphique remplit avec l'ensemble associés au numéro logique.

Ces paramètres sont, soit ceux définis par l'utilisateur, soit ceux pris par défaut par le système. Ce sont notamment les dimensions de la fenêtre dans l'espace utilisateur et de la clôture dans l'espace de visualisation, les valeurs des différents attributs de tracé, etc...

6.8.3.8. Lecture de l'attribut couleur d'un point écran

Cette fonction ne peut être exécutée que sur les environnements qui ont la capacité de relire l'attribut couleur d'un point écran.

Le point précisé est en coordonnées utilisateur et celles-ci sont données dans l'ordre X, Y. L'ensemble fenêtre/clôture associé au numéro logique permet au système graphique de calculer les coordonnées réelles (coordonnées écran) correspondantes.

Le système retourne la valeur de l'attribut couleur du point écran correspondant au point précisé. Ce dernier doit être situé à l'intérieur de la fenêtre associée au numéro logique. Sinon une erreur est retournée à l'utilisateur.

6.8.4. Fonctionnement du système graphique

6.8.4.1. Organisation système

Dans l'organisation du système graphique, on distingue deux niveaux :

- Le noyau graphique, partie logique qui assure, de façon indépendante des particularités des différents environnements, les fonctionnalités offertes à l'utilisateur.
- Le contrôleur périphérique qui comprend toute la partie interaction avec la machine.

Un interface précis avec des fonctions générales et indépendantes des caractéristiques des environnements a été défini entre le noyau graphique et le contrôleur périphérique (fonctions de type POINT, VECTEUR, ARC DE CERCLE,...).

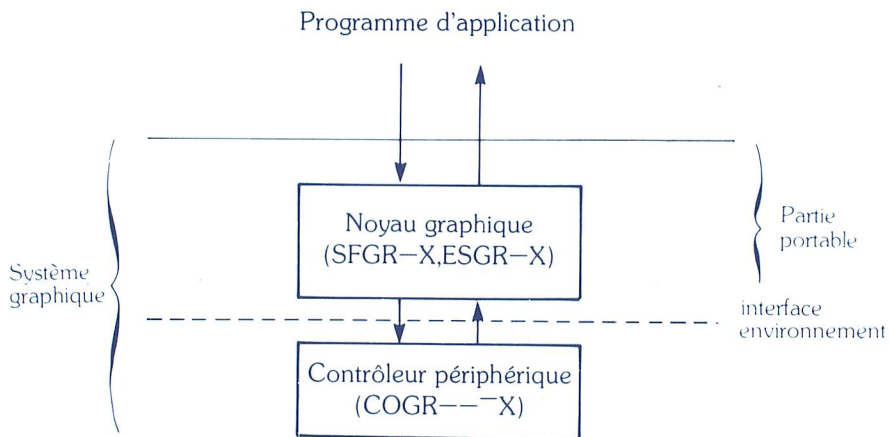
Entre deux machines différentes, seul le contrôleur périphérique change (voir schéma 8-K).

6.8.4.2. Portabilité

Afin d'assurer la portabilité des applications graphiques, le système ne doit pas rejeter en erreur celles-ci lorsqu'elles font appel à une fonctionnalité graphique qui n'existe pas dans l'environnement présent.

Pour qu'une même application puisse être exécutée sur des machines différentes, il doit donc y avoir une « adaptation » du système graphique à l'environnement présent.

Pour cela, a été défini, au niveau du contrôleur, un descripteur qui permet de préciser les caractéristiques du périphérique, notamment le nombre de points écran adressables (pixels), le nombre de couleurs supportées, les différentes possibilités de tracé (POINT? VECTEUR? ARC DE CERCLE?...). Chaque contrôleur connecté possède un descripteur de ce type, rempli avec ses propres caractéristiques. Le noyau graphique peut consulter ce descripteur et « s'adapter » ainsi à chaque environnement.



ORGANISATION DU SYSTEME GRAPHIQUE

Schéma 8-K

6.8.4.3. Niveau minimum de fonctionnalités graphiques

Afin d'offrir à l'utilisateur un certain nombre de fonctionnalités, quelque soit l'environnement, le noyau graphique en simule quelques-unes. Par exemple, il effectue point à point, lorsque ceux-ci n'existent pas dans l'environnement présent, les six premiers types de symbole avec les trois premières tailles et les quatre premiers types de ligne avec les trois premières épaisseurs. Cela permet à l'utilisateur d'avoir toujours un minimum de fonctions graphiques, y compris sur des environnements « pauvres ».

6.8.4.4. Principe de fonctionnement

Pour des raisons de temps de réponse et d'efficacité et afin de ne pas pénaliser les environnements « riches », le noyau graphique essaie toujours d'exploiter au mieux les possibilités de l'environnement.

Ainsi, pour chaque fonctionnalité graphique demandée, le noyau graphique vérifie dans le descripteur du contrôleur périphérique si celle-ci existe dans l'environnement, auquel cas il la lui transmet. Sinon, s'il le peut, le noyau la simule. Enfin, en dernier recours, il adopte une solution de rechange proche et disponible, soit dans l'environnement, soit par simulation.

Par exemple, pour une épaisseur de trait demandée, ni disponible dans l'environnement ni par simulation, le système graphique la remplace par l'épaisseur la plus proche réalisable, soit par l'environnement, soit par simulation.

Une application qui a été écrite pour être exécutée sur un environnement « riche », avec un grand nombre de couleurs par exemple, peut ainsi fonctionner sur un environnement plus « pauvre », disposant de moins de couleurs. L'aspect du dessin visualisé ainsi que le temps de réponse sont différents d'une machine à l'autre.

6.9. LES TRANSMISSIONS

6.9.1. La procédure PTTY

6.9.1.1. Introduction

PRESENTATION

La procédure PTTY développée sous PROLOGUE 16 bits permet d'utiliser les voies V24 en mode ASYNCHRONE.

La PTTY est accessible à l'utilisateur à travers l'interface du système de fichiers.

La procédure PTTY doit être déclarée dans le fichier SYSCONF pour pouvoir être utilisée. Son code mnémonique de ressource fichier est "LA".

Le nombre de voies V24 utilisables simultanément par différents utilisateurs est de 4.

L'utilisateur indique la voie V24 qu'il veut utiliser, en fournissant un numéro d'unité compris entre 0 et 3.

(Ex. LA0 => Unité 0 asynchrone)

L'affectation des voies V24 à la PTTY est faite à l'aide du fichier SYSCONF. Il est possible de définir 4 unités asynchrones. A chaque unité est associée une voie V24.

Les possibilités offertes par la procédure PTTY sont les suivantes :

- 5, 6, 7 ou 8 bits par caractère
- Parité paire ou impaire ou sans parité
- 1, 1.5 ou 2 stop bits par caractère
- Vitesses de 50 à 19 200 bauds
- BREAK en émission ou en réception
- Gestion des signaux modem
- Horloge interne

Les fonctions réalisées par la procédure sont les suivantes :

- Réservation procédure (OPEN)
- Connexion
- Emission de données
- Emission avec écho
- Réception de données
- Envoi d'un BREAK
- Déconnexion
- Libération procédure (CLOSE)

6.9.1.2. La configuration de la procédure PTTY

La procédure de transmission asynchrone "PTTY" se compose de deux modules d'extension "PTTY-X" et "TTYxx-X" (où "xx" dépend de votre environnement), qui se chargent au moyen de l'utilitaire de configuration "SYSCONF-X". Les modules doivent alors être déclarés comme suit, dans le fichier "SYSCONF-S" (pour plus de détails, reportez-vous au chapitre 7 "Configuration de PROLOGUE") :

- PTTY < SFX (P1=20H,P2=0)
 < ES (P1='LA',P2=4,P3='PTTY')
- TTYxx (P1=v) < PTTY

où "v" représente le numéro de la voie V24 sur lequel doit avoir lieu la transmission.

Remarque :

On peut vérifier la présence de la procédure "PTTY" par VPMI.

Exemple :

```
-> VPMI<rc>  
    Visualisation et modification des implicites de poste  
    Version 1.2a Copyright (c) 1984 BULL Micral
```

PROLOGUE V2.3

périphériques : MD FL IM LA

nombre d'unités : 1 1 1 4

implicite programme : MD0

implicite fichier : MD0

implicite imprimante : IM0

Modifications :

implicite programme :

Le mnémonique “LA” indique la présence de la procédure de transmission “PTTY”.

– Si les deux systèmes — émetteur et récepteur — fonctionnent sous PROLOGUE 2.3., ils doivent avoir déclaré la procédure “PTTY” dans leur système. Par contre, ils ne sont pas obligés de faire leur transmission à partir de la même voie V24 (la valeur du paramètre “v” peut donc être différente).

6.9.1.3. La commande de transmission de fichiers : “TELE”

Le lancement de la transmission d'un fichier se fait au moyen de l'utilitaire TELE-X activé d'ABORD sur le site récepteur, PUIS sur le site émetteur. Une fois la transmission des données terminée, on reviendra à l'interpréteur. On comparera alors le nombre de blocs émis et celui reçu, afin de voir si aucun bloc n'a été perdu. Il est possible d'interrompre une transmission en frappant sur un site la touche “ESC”. Dans ce cas, il y a retour à l'interpréteur de commandes avec affichage préalable du message “*** Abandon ***”. Il faudra alors frapper aussi la touche “ESC” sur l'autre site pour débloquer ce dernier.

Syntaxe pour l'émetteur :

—> [unité.]TELE,desfic,EM[,vitesse]<rc>

Syntaxe pour le récepteur :

—> [unité.]TELE,desfic,RC[,vitesse]<rc>

où :

unité	Indique l'unité sur laquelle l'utilitaire TELE est recherché afin d'être chargé en mémoire. Si l'unité est omise, l'utilitaire est recherché sur l'unité implicite programme de votre poste.
TELE	Nom du fichier binaire correspondant, de suffixe “X”.
desfic	Descripteur du fichier à transmettre (première syntaxe) ou du fichier de réception (deuxième syntaxe). Si le type n'est pas spécifié, c'est le type “S” qui est pris par défaut.

EM	Mot-clé indiquant qu'on désire lancer la transmission pour émettre des données.
RC	Mot-clé indiquant qu'on désire lancer la transmission pour recevoir des données.
vitesse	Vitesse de transmission entre les deux systèmes. Doit être exprimée en bauds et spécifiée en décimal. Par défaut, c'est 9 600 bauds.

Remarques :

- Le fichier d'émission et le fichier de réception ne portent pas forcément le même nom ; c'est le contenu que l'on transfère, pas les descripteurs.
- Si la vitesse est spécifiée, elle doit être la même dans la commande d'émission et dans la commande de réception.
- Le suffixe "S" étant pris par défaut, pour transmettre un fichier sans suffixe, il conviendra, soit de dupliquer celui-ci en spécifiant un suffixe, soit le renommer, toujours en spécifiant un suffixe.
- Sur le site de réception, si le fichier devant contenir les données n'existe pas, il sera créé. Si l'utilisateur décide d'abandonner la transmission, il sera détruit.

MESSAGES D'EXECUTION

Une fois la commande lancée, l'utilitaire TELE affiche

"Nb. de Blocs transférés : 0"

puis émet des signaux sonores jusqu'à ce que la transmission commence. A partir de ce moment, à chaque bloc émis ou reçu, le compteur de blocs transmis est mis à jour. Lorsque la transmission est terminée, les deux sites affichent un "—" puis reviennent à l'interpréteur de commandes. Le nombre de blocs transférés doit alors être identique sur les deux sites et doit correspondre à la taille du fichier (en secteurs).

Exemple :

—> /,FICH01

Catalogue du volume Version 3.4b

Copyright (c) 1984 BULL Micral

Vol: KLEMANN date: 4.11.1984 Nb fic: 895/235 Lg. Gr: 32 Occup: 1374/806

FICH01-S : 19.11.1984 1 : (700, 700)

—> TELE, FICH01-S, EM Nb. de Blocs transférés : 32

—> _

MESSAGES D'ERREUR

En plus des codes erreurs système dont vous trouverez la signification dans l'annexe B, TELE affiche quelques messages propres.

Lorsqu'une erreur dans la commande a été détectée, TELE rappelle la syntaxe d'appel.

Exemple :

—> TELE, FICH01-S, ER, 1200

Erreur de Commande n° 30 Syntaxe Incorrecte

=> TELE, nom-fichier, sens[, vitesse]

exemple : TELE, FL0.FICH01, EM (vitesse par défaut : 9 600)

OU TELE, FICH1, RC, 1200

(TELE utilise la procédure PTTY - unité 0)

—> _

* “Fichier vide”

Ce message est affiché lorsque TELE a trouvé le fichier dans le catalogue, mais que ce fichier n'a aucun espace alloué.

Exemple :

—> CP, CF, FICH01-S _

—> TELE, FICH01, EM, 1200

Fichier vide

—>

Le fichier "FICH01-S" (on rappelle que le suffixe "S" est pris par défaut) a bien été trouvé, mais venant d'être créé, il est vide.

*** ..*** Abandon ***.,**

Ce message est affiché lorsque l'utilisateur a abandonné la transmission d'un fichier, en frappant la touche ESC.

Exemple :

```
-> TELE,SYSCONF-S,EM Nb. de Blocs transférés : 3
      *** Abandon ***
-> _
```

La transmission du fichier "SYSCONF-S" a été abandonnée après l'émission du troisième bloc.

*** "Voie asynchrone occupée"**

Ce message apparaît lorsque la voie, utilisée pour faire la transmission, est déjà occupée, soit par une transmission en cours (dans ce cas, il faut attendre la fin de la transmission), soit par un poste qui utilise cette voie V24 (dans ce cas, il convient de reconfigurer le système de manière à ce que la voie soit libre).

Exemple :

```
-> TELE,SYSCONF-S,EM
      VOie asynchrone occupée
-> _
```

*** "Procédure PTTY (unité 0) non déclarée (voir sysconf)"**

Ce message apparaît lorsqu'on souhaite faire une transmission par TELE et que la procédure PTTY n'existe pas dans le système. Il convient alors de modifier la configuration, recharger le système puis relancer l'opération.

Exemple :

—> TELE,FICH01,EM

Procédure PTTY (unité 0) non déclarée (voir sysconf)

—> VPML

Visualisation et modification des implicites de poste

Version 1.2a Copyright (c) 1984 BULL Micral

PROLOGUE V2.2

périphériques : MD FL IM

nombre d'unités : 1 1 1

implicite programme : MD0

implicite fichier : MD0

implicite imprimante : IM0

Modifications :

implicite programme :

La commande TELE a été refusée, la procédure PTTY étant absente de la configuration. Ceci est vérifié par VPML qui ne nous indique pas la présence de la ressource périphérique "LA" (Ligne Asynchrone).

6.9.2. La procédure B.S.C. 2780 - 3780 - 3741

6.9.2.1. Introduction

La procédure B.S.C. (Binary Synchronous Communication), développée sous Prologue permet la transmission de données suivant les standards B.S.C. 2780, 3780 et 3741.

PRESENTATION

Toute communication a pour but la transmission d'une certaine quantité d'informations. Elle implique l'intervention d'un transmetteur (émetteur), d'un destinataire (récepteur) et d'une liaison de communication (voie de transmission du message).

La communication à distance est le télécommunication.

Les communications entre ordinateurs sont des transferts d'informations. Il existe plusieurs modes pour transférer des informations : asynchrone, synchrone, bit (HDLC, SDLC) ...

La procédure B.S.C. (Binary Synchronous Communication) définit un ensemble de règles qui permettent de gérer une transmission synchrone de données codées binaires entre deux unités centrales éloignées l'une de l'autre.

“Synchronous Communication” signifie qu'après reconnaissance d'une séquence spéciale de synchronisation, le récepteur est synchronisé avec l'émetteur pendant toute la durée d'une transmission.

La procédure B.S.C. permet à la totalité d'une transmission de s'effectuer automatiquement et correctement grâce à une gestion des erreurs de transmission.

Une liaison de communication se compose d'équipements de communications au travers desquels deux stations (ordinateurs) peuvent communiquer :

- Logiciels (codages, contrôles, corrections)
- Cordons V24
- Lignes téléphoniques
- MoDem (Modulateur-Démodulateur de signaux électriques) ...

La procédure B.S.C. gère une liaison dite point à point, c'est-à-dire que seules deux stations utilisent une ligne de communication.

CODAGE DE L'INFORMATION

CARACTERES ET SEQUENCES DE CONTROLES

La transmission de caractères étant séquentielle, des caractères de contrôles sont intercalés avec des caractères de données pendant la transmission entre deux stations.

Les caractères de contrôles servent à identifier et délimiter les messages échangés entre deux stations.

Les séquences de contrôles sont des messages ne comportant pas de données de l'utilisateur. Elles sont destinées à assurer la synchronisation entre stations, ainsi que la régulation et le contrôle du bon acheminement des messages de données.

L'utilisation de ces caractères et séquences de contrôles est définie par le protocole B.S.C.

MODE TRANSPARENT

Il se peut que les données à transmettre aient la même codification que certains codes de contrôles. Dans ce cas, le mode transparent permet la transmission de données indépendamment de leurs codes de représentation.

Un mode transparent, les codes de contrôles sont précédés d'un caractère spécial : ("Data Link Escape").

Le mode transparent permet donc la transmission de données codées en binaire sur 8 bits.

Le code généralement utilisé est le code EBCDIC de chez IBM.

LE MODE 3780

Le mode 3780 traite des blocs de données en mode non transparent.

Un bloc de données ne doit pas être découpé en sous-blocs (STX ..ETX).

En mode 3780, il n'y a jamais de transmission d'une marque de fin de fichier (STX EOT).

En mode 3780, il n'y a jamais de demande de fin de transmission (EXT).

LE MODE 3741

Le mode 3741 est lié au format 3740 des disquettes 8 pouces de chez IBM.

Une disquette au format 3740 contient 77 pistes de 26 secteurs de 128 octets.

LES BLOCS DE DONNEES AVEC LE MODE 3741

Un bloc de données contient au plus 128 octets significatifs, sans les caractères de contrôles.

Un enregistrement peut être composé d'un ou plusieurs blocs de données, il se termine toujours par bloc de données vide, de la forme STX ETX.

6.9.2.2. Configuration de la procédure B.S.C. 2780 - 3780 - 3741

LE PARAMETRAGE DES CONTROLEURS

• POUR LES PROCEDURES B.S.C. 2780, 3780, 3741

A la configuration du système, il faut définir les paramètres suivants :

- P1 = Numéro de voie physique associée à l'unité 0
- P2 = Vitesse de transmission (= 0 avec un modem)
- P3 = Nombre de caractères de synchronisation
- P4 = Nombre de caractères de remplissage (PAD)
- P5 = Temps de retournement de la ligne

P1 et P2 sont obligatoires, P3, P4 et P5 sont optionnels.

Les valeurs par défaut des paramètres sont :

P3=4 , P4=2 , P5=0 .

P1 peut prendre les valeurs 1 ou 2.

P2 = 0 sélectionne, comme horloge de transmission, le signal présent sur la patte 24 du connecteur cannon 25 points (mode externe).

P2 ≠ 0 précise la valeur de la vitesse de transmission et sélectionne le mode horloge interne.

Si le calculateur avec lequel l'utilisateur souhaite dialoguer a du mal à se synchroniser, il faut augmenter le nombre de caractères de synchronisation (paramètre P3).

Si le calculateur avec lequel l'utilisateur souhaite dialoguer reçoit des messages incomplets, il faut augmenter le nombre de caractères de remplissage (paramètre P4).

Le temps de retournement de la ligne correspond à la temporisation à effectuer avant d'émettre. Ce temps est exprimé en centième de seconde. Une mauvaise valeur peut ralentir la transmission des messages.

L'utilisation de ce paramètre n'est pas conseillée.

La B.S.C. étant déclarée dans le fichier SYSCONF-S, il faut déclarer également le contrôleur de ligne BSCxx-X.

Syntaxe de déclaration

Exemple :

Initialisation de l'unité 0 sur la voie 1 (adr. 3A0), à 1200 bauds avec 6 caractères de synchronisation, 3 caractères de remplissage et un temps de retournement de 100 millisecondes.

- BSC < SFX (P1=20h,P2=0)
 < ES (P1='LS',P2=1,P3='BSC')
- BSCxx (P1=1,P2=1200,P3=6,P4=3,P5=10) < BSC

6.9.3. La procédure B.S.C. 3270

6.9.3.1. Introduction

La procédure B.S.C. 3270 sous PROLOGUE assure le transport de données codées dans une présentation 3270 sur une ligne B.S.C. multipoint.

La procédure B.S.C. 3270 permet de connecter 32 terminaux (devices) sur une ligne B.S.C. Le logiciel émule une grappe terminale et permet les échanges de trames B.S.C. entre un central et des terminaux connectés à une grappe appelée unité de contrôle (Control Unit).

Cette procédure n'est en aucun cas une simulation interactive d'un terminal écran ou imprimante IBM 32xx.

Il n'y a donc pas de gestion de protocole de "bout en bout", c'est-à-dire que les données émises ou reçues ne sont pas interprétées (sauf le WCC des Write Command).

Seul le protocole ligne de transmission est géré, ce qui inclut la structuration des blocs de données, le calcul du BCC (Binary Control Check), les répétitions en cas d'erreur, les temporisations des transmissions, etc...

CARACTERISTIQUES

TRANSMISSION : Synchrone half duplex.
Protocole B.S.C. ASCII ou EBCDIC
Gestion du mode de données transparent.

LIAISON : Multipoint (polling / selecting).
Grappe 3271, 3274 ou 3276.
32 terminaux (devices) au plus..

LOGICIEL : Système d'exploitation PROLOGUE.
Intégré à TCS.

Le logiciel BSC 3270 occupe environ 25 K octets (SFTP-X et BSC3270-X) en mémoire avec le contrôleur de ligne synchrone B.S.C. pour un écran connecté.

Paramétrage de la procédure à l'établissement de la connexion : type du matériel émulé, 3271, 3274 ou 3276, adresses, Time-Out et paramètres de la ligne synchrone ...

Accès : Interface avec le mécanisme SF de PROLOGUE. Instructions de gestion de fichiers du langage B.A.L.

6.9.3.2. Configuration de la procédure B.S.C. 3270

B.S.C. 3270 est considéré comme une extension du système d'exploitation PROLOGUE de base.

Tous les modules qui le constituent sont automatiquement intégrés au système lors de la phase de chargement et deviennent résidents en mémoire. Pour ce faire, chaque module se présente sous la forme d'un fichier binaire présent sur le disque système lors du chargement.

B.S.C. 3270 est également déclaré dans le fichier SYSCONF-S

SYSCONF-S est un fichier source décrivant les modules que l'on désire utiliser, en extension à PROLOGUE et des liens que comportent ces modules entre eux et vis-à-vis de PROLOGUE.

La figure suivante représente par des flèches les liens de configuration qu'il convient d'établir pour que l'intégration de B.S.C. 3270 PROLOGUE soit correcte.

Le paramètre P1 de SFTP définit la taille mémoire exprimée en K octets, utilisable par l'ensemble des protocoles pour obtenir les descripteurs nécessaires à l'établissement des liaisons ainsi que les tampons contenant les messages échangés. Par défaut, P1 est égal à 32 K. Sinon, la valeur de P1 peut varier de 4 à 256 K par valeur paire.

Le mécanisme d'allocation de la mémoire fonctionne de la façon suivante : SFTP alloue des blocs de 2 K en commençant par un seul bloc. S'il est nécessaire de disposer de plus de mémoire, des blocs supplémentaires sont successivement alloués tant que la valeur spécifiée par P1 n'est pas atteinte.

Un paramètre P1 trop restrictif peut produire une erreur "manque de mémoire" lors de l'établissement d'une nouvelle liaison, ou provoquer des répétitions de message en ligne à la réception lorsque celle-ci est établie.

Le calcul de l'espace mémoire nécessaire pour une utilisation donnée est complexe à réaliser. Les paramètres qui entrent en jeu sont nombreux en première approximation, il est bon de prévoir 5 K par device connecté.

6.9.4. La procédure VIP/PVE

6.9.4.1. Introduction

La procédure VIP/PVE, développée sous PROLOGUE permet la transmission de données suivant les standards VIP.

PRESENTATION

Toute communication a pour but la transmission d'une certaine quantité d'informations. Elle implique l'intervention d'un transmetteur (émetteur), d'un destinataire (récepteur) et d'une liaison de communication (voie de transmission du message).

La communication à distance est la télécommunication.

Les communications entre ordinateurs sont des transferts d'informations. Il existe plusieurs modes pour transférer des informations : asynchrone, synchrone, bit (HDLC, SDLC) ...

La procédure VIP définit un ensemble de règles qui permettent de gérer une transmission synchrone de données codées binaires entre deux unités centrales éloignées l'une de l'autre.

Synchrone signifie qu'après reconnaissance d'une séquence spéciale de synchronisation, le récepteur est synchronisé avec l'émetteur pendant toute la durée d'une transmission.

La procédure VIP permet à la totalité d'une transmission de s'effectuer automatiquement et correctement grâce à une gestion des erreurs de transmission.

Une liaison de communication se compose d'équipements de communications au travers desquels deux stations (ordinateurs) peuvent communiquer :

- Logiciels (codages, contrôles, corrections)
- Cordons V24
- Lignes téléphoniques
- MoDem (Modulateur-Démodulateur de signaux électriques) ...

La procédure VIP gère une liaison dite au point à point, c'est-à-dire que seules deux stations utilisent une ligne de communication.

6.9.4.2. Configuration de la procédure VIP/PVE

A la configuration du système, il faut définir, dans SYSCONF-S, les paramètres suivants :

- P1 = Numéro de voie physique associée à l'unité 0
- P2 = Vitesse de transmission (= 0 avec modem)
- P3 = Nombre de caractères de synchronisation (SYN)
- P4 = Nombre de caractères de remplissage (PAD)
- P5 = Temps de retournement de la ligne

P1 et P2 sont obligatoires, P3, P4 et P5 sont facultatifs.

Les valeurs par défaut de ces paramètres sont :

P3=4 ,P4=2 ,P5=0

Syntaxe de la déclaration

Exemple :

Initialisation de l'unité 0 sur la voie physique 2 (adr. 380)

```
VIP      < SFX (P1=20h,P2=0)
          < ES  (P1='LV' ,P2=1,P3='VIP')
VIPxx (P1=2,P2=0)    < VIP
```


6.9.5. La procédure X25

6.9.5.1. Introduction

Sous prologue, BULL MICRAL offre une procédure de transmission permettant de se raccorder à un réseau de transport utilisant une technique de commutation de paquets conforme à la norme X25 (exemple = TRANSPAC).

Le mécanisme de cette procédure est réalisée par deux modules logiciels dénommés :

X25-X
LAPB-X

dont la description, l'exploitation, et l'intégration dans un programme sont décrites dans les manuels cités en Annexe

Le fonctionnement de cette procédure sur un ordinateur personnel BULL MICRAL nécessite l'utilisation d'un module logiciel (le contrôleur de ligne) :

HLAPxx-X

qui réalise une adaptation au matériel. Le présent chapitre traite de la mise en œuvre de ce module.

L'ADAPTATEUR DE LIGNE HDLC/SDLC

Au plan matériel, le contrôleur de ligne HLAPxx-X interface avec l'adaptateur de ligne HDLC/SDLC.

Bien que HLAPxx-X ne gère qu'une liaison logique (unité 0) à la fois, il est néanmoins possible d'équiper l'ordinateur personnel BULL MICRAL de deux adaptateurs de ligne HDLC/SDLC au plus, à condition de les doter d'adresses différentes.

6.9.5.2. La configuration de la procédure X25

La syntaxe générale de cette déclaration est :

HLAP_{xx} (P1 = i) < LAPB
 < SFTP

le paramètre P1 est obligatoire.

En fonction du micro-ordinateur il peut prendre les valeurs 1 ou 2 suivant le choix de la voie effectué.

Exemple :

Voici un exemple de déclarations que doit contenir le fichier SYSCONF-S dans le cas d'une liaison utilisant la procédure X25.

– SFTP (P1 = X) < SFX (P1=20h,P2=0)
 < ES (P1="TP",P2=1,P3="SFTP")

– X25 < SFTP

– LAPB < SFTP

– HLAP_{xx} (P1 = 1) < LAPB
 < SFTP

– CFTCS

Remarque :

X est la taille de la zone des tampons de SFTP, c'est un multiple de 2K octets.

CFTCS est le configurateur du chemin, il utilise un fichier de données CFTCS-S qui contient la description des différents chemins possibles.

6.10. LES RESEAUX

6.10.1. Généralités

Un réseau de télécommunications est l'ensemble des moyens mis en œuvre pour permettre à des usagers distants d'échanger entre eux des informations.

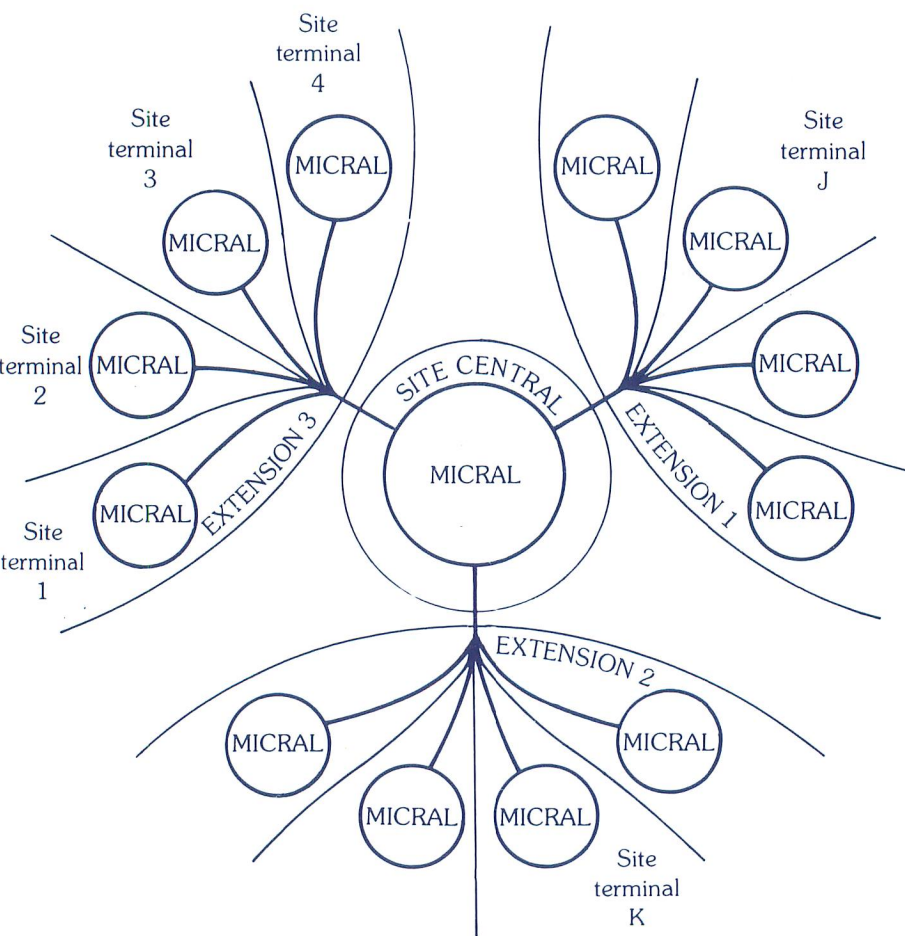
En France et dans la plupart des pays européens, les télécommunications constituent un monopole d'Etat. Pour cette raison, les réseaux de transmission de données à accès "ouvert", sont des réseaux publics.

Tenant compte des besoins importants de communication de données exprimés par les utilisateurs, les différentes administrations concernées (les PTT en France), ont réfléchi à la mise en service d'un nouveau service de transport répondant aux besoins de toutes les catégories d'utilisateurs :

- applications conversationnelles (interrogation de fichiers, temps partagé, gestion de transactions ...)
- soumission de travaux à distance
- interconnexion de calculateurs (transferts de fichiers...)
- messagerie, traitement de textes...

6.10.2. Le réseau SPR

Le système de Partage de Ressources “SPR” permet, à travers un réseau “étoilé”, de connecter de un à douze micro-ordinateurs à un site central dont ils peuvent partager les ressources.



6.10.2.1. Introduction

DESCRIPTION GENERALE

Chaque MICRAL Terminal est relié par une liaison point à point au MICRAL Serveur indépendamment des autres ordinateurs du réseau.

La mise en œuvre de SPR gère la liaison entre le MICRAL Serveur et une MICRAL Terminal.

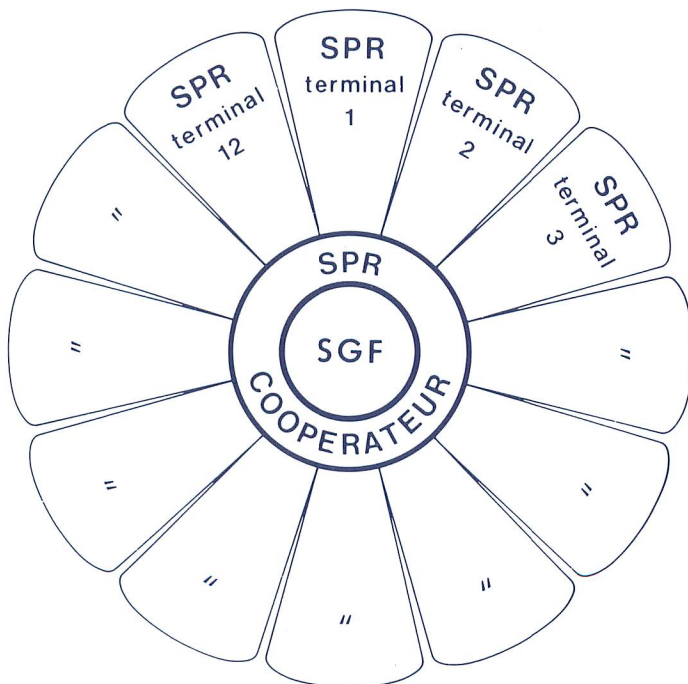
Tout MICRAL Terminal peut néanmoins travailler indépendamment du reste du réseau comme un micro-ordinateur autonome.

Sous SPR, le MICRAL Serveur coordonne les demandes adressées par les différents MICRAL Terminals.

Dans la suite de ce document le MICRAL Serveur est appelé "coopérateur" ou "central". Les MICRAL Terminals reliés au coopérateur par la liaison SPR sont appelés "TERMINAUX".

Un MICRAL Terminal adresse une demande appelée "requête" au coopérateur. Cette requête lui permet l'accès aux ressources du MICRAL Serveur (mémoire de masse, fichiers, système de télécommunication, imprimante...) et reçoit des réponses.

Tous les utilisateurs peuvent ainsi disposer des ressources regroupées sur le site central autour du MICRAL Serveur.



GENERALITES MATERIEL ET LOGICIEL

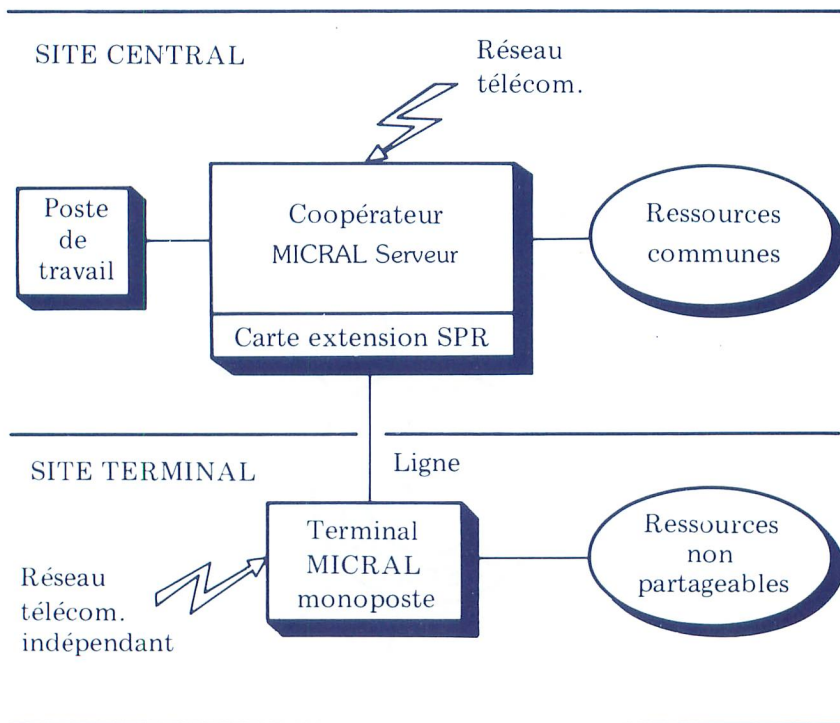
Généralités matériel

Le MICRAL serveur pour jouer le rôle de coopérateur, doit être muni d'une ou plusieurs cartes extensions SPR (3 maximum). Les cartes sont reliées au boîtier par un cordon.

Le MICRAL terminal possède en standard une prise SPR, et peut toujours servir de terminal.

Généralités logiciel

Le logiciel SPR est un ensemble de modules (extension de PROLOGUE) dont certains sont dans le système coopérateur, les autres dans les systèmes terminaux. SPR travaille avec le système de gestion de fichiers SGF de PROLOGUE (à partir de la version 2.2 de PROLOGUE).



6.10.2.2. La configuration du réseau SPR

La disquette SPR comprend tous les modules nécessaires pour pouvoir configurer le système adapté à l'utilisateur. Ces modules, fournis au système PROLOGUE, seront automatiquement isérés dans le système utilisateur, lors de la configuration.

La configuration d'un ensemble travaillant sous SPR est faite en deux temps :

- Configuration du coopérateur
- configuration de chaque terminal.

Pour plus de détails, référez-vous au Guide Utilisateur de SPR.

6.11. L'option "Multivue"

6.11.1. Introduction

PROLOGUE MULTIVUE permet à un même utilisateur sous PROLOGUE, d'exécuter SIMULTANEMENT, plusieurs programmes à partir de la même console. Chaque programme travaille dans son propre écran, indépendamment des autres. Un des écrans, dit écran actif, est effectivement affiché sur l'écran physique. Les autres écrans sont entretenus en mémoire, permettant un fonctionnement des programmes en tâches de fond.

La fonctionnalité MULTIVUE peut être étendue à plusieurs consoles.

L'option MULTIVUE ne demande aucune modification à apporter aux programmes et utilitaires fonctionnant sous le système PROLOGUE. Il en est de même pour le système PROLOGUE (version 2.3), à l'exception des accès directs au matériel.

Pour obtenir cette fonctionnalité, il suffit à l'utilisateur d'ajouter à son système déjà existant, un module standard COVS et un ou plusieurs contrôleurs écrans (adaptés MULTIVUE ou non) qu'il déclare dans le fichier SYSCONF-S.

Sur une console physique, il ne peut y avoir au maximum que 4 tâches interpréteur, c'est-à-dire 4 programmes actifs, simultanément, le nombre de tâches interpréteur que sait gérer le système étant au maximum de 8, pour l'ensemble des utilisateurs.

Sur la console physique, le passage d'une tâche à une autre tâche se fait, soit à l'aide du clavier, soit à l'aide d'un utilitaire.

Le système MULTIVUE utilise la vingt-cinquième ligne de la console physique, pour informer l'utilisateur de son état de fonctionnement. Si cette ligne est déjà utilisée par des programmes d'application, le système MULTIVUE peut être paramétré pour conserver cet état.

Pour illustrer Multivue, nous pourrions comparer le concept des tâches à des chaînes de télévision dont les programmes se déroulent simultanément.

En télévision, l'affichage d'une chaîne se fait par un commutateur ; en fonction MULTIVUE, le changement d'écran se fait par une combinaison de touches.

6.11.2. Fonctionnement

6.11.2.1. Mode général

Au changement du système PROLOGUE, une tâche interpréteur est allouée par console physique.

Au départ, le système est donc “monotâche” sur chaque console physique. Au fur et à mesure des besoins de l'utilisateur et des capacités du système, l'utilisateur peut disposer sur sa console physique, au maximum de 4 tâches interpréteur. Pour cela, il doit frapper au clavier une combinaison de touches spéciales, définie par configuration.

Le nombre total des tâches interpréteur est celui paramétré dans le système (nombre de postes dans le configurateur).

Chaque tâche interpréteur possède ses propres paramètres systèmes (implicite fichier, implicite programme, touches fonction, touches programmables, définition de l'écran). Il en va de même pour l'utilisation de la vingt-cinquième ligne.

6.11.2.2. Mode graphique

Une tâche interpréteur doit occuper l'écran de la console physique, au moment où elle utilise le mode graphique.

Sinon, elle est bloquée par le contrôleur au moment du passage en mode graphique. Elle n'est débloquée qu'à l'instant où elle occupe à nouveau l'écran de la console.

6.11.2.3. Mode transparent

Les accès écrans en mode transparent (ESC T...) sont filtrés par Multivue.

6.11.3 L'utilitaire "POSTE"

Cet utilitaire permet l'allocation ou la commutation d'une tâche interpréteur, la désallocation d'une tâche interpréteur et l'envoi d'une commande à une autre tâche interpréteur.

Pour être utilisées, les différentes tâches doivent être activées par le clavier.

6.11.3.1. L'allocation ou la commutation

SYNTAXE DE LA COMMANDE :

—> [unité.]POSTE,AC,ntâche<rc>

avec AC = mot-clé de l'allocation d'une tâche interpréteur ou de la commutation vers une autre tâche interpréteur qui doit déjà avoir été activée au clavier.

ntâche = le numéro de la tâche interpréteur (de 1 à 4).

L'allocation se fait à partir de n'importe quelle tâche.

Mais le nombre "ntâche" doit être immédiatement supérieur à celui de la dernière tâche interpréteur allouée pour cette console physique. Sinon, la commande est ignorée.

Après cette opération, la tâche allouée voit son écran devenir actif.

La commutation permet, à partir de la tâche disposant de l'écran physique, de visualiser une autre tâche interpréteur (cette tâche a de ce fait, son écran actif).

6.11.3.2. La désallocation

SYNTAXE DE LA COMMANDE :

—> [unité.]POSTE,DL<rc>

avec DL = mot-clé de la désallocation d'une tâche interpréteur.

La désallocation ne peut avoir lieu que pour la dernière tâche allouée sur la console concernée.

C'est la tâche elle-même qui demande à se désallouer, et uniquement quand elle est au repos (à la flèche de l'interpréteur de commande).

Après la désallocation, la tâche de numéro immédiatement inférieur se voit attribuer l'écran physique.

6.11.3.3. L'envoi de caractères

SYNTAXE DE LA COMMANDE :

—> [unité.]POSTE,EM,ntâche,caractères<rc>

avec EM = mot-clé de l'envoi de caractères vers une autre tâche interpréteur.

ntâche = le numéro de la tâche interpréteur vers laquelle ces caractères sont envoyés.

caractères = représente les caractères à envoyer ; le caractère de validation pour une commande <rc> doit être remplacé par ^B.

L'envoi de caractères peut être fait à partir d'un programme ou à partir de l'interpréteur de commande d'une autre tâche.

La tâche réceptrice verra ces caractères tout comme s'ils avaient été frappés au clavier lorsqu'elle a la disposition de l'écran physique.

6.11.4. Présentation de la vingt-cinquième ligne système

La vingt-cinquième ligne se divise en deux parties :

- La partie système
- La partie MULTIVUE

La partie système occupe la moitié droite de cette ligne

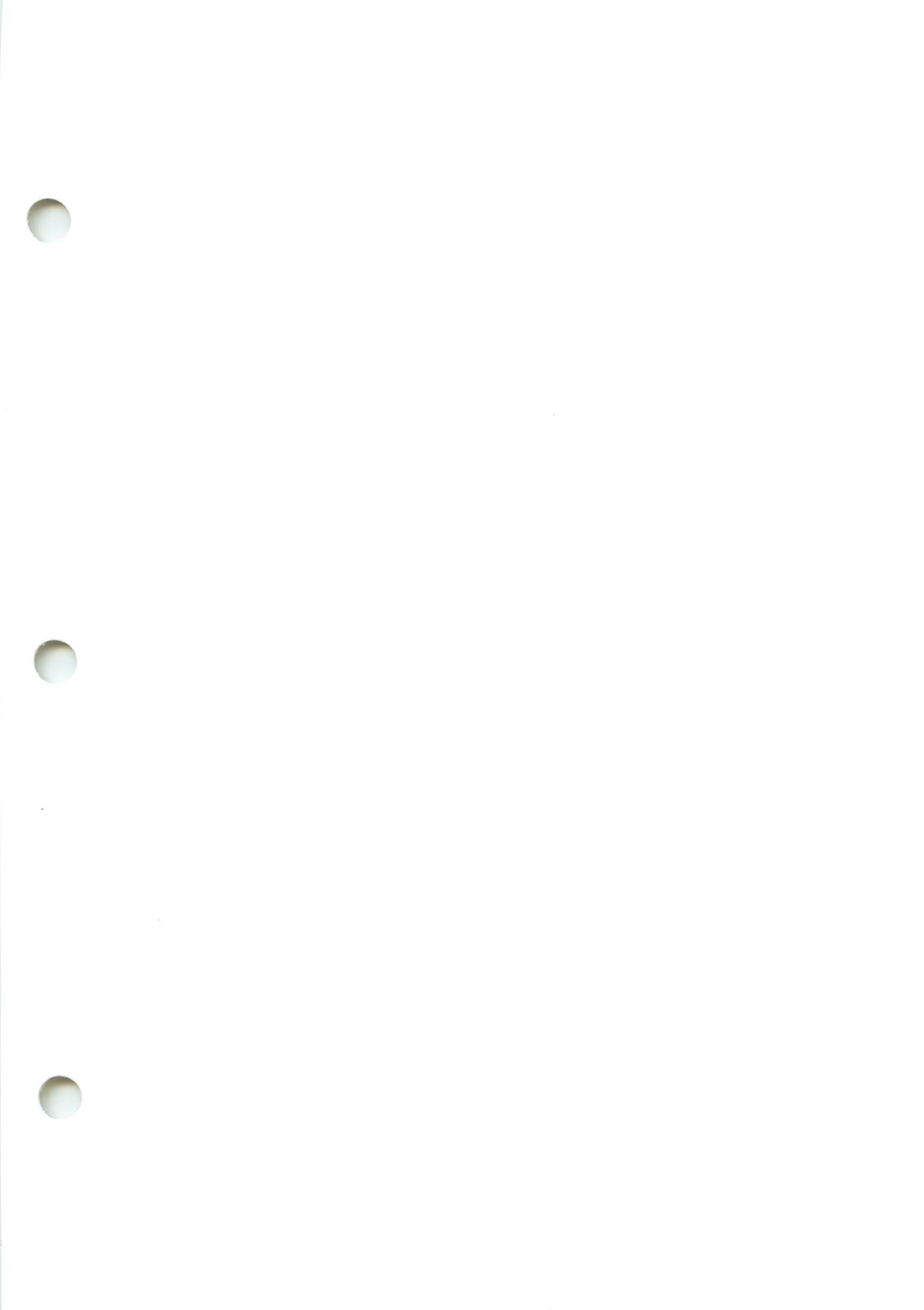
La partie MULTIVUE, la moitié gauche.

L'heure et les renseignements claviers sont affichés dans la partie système. Les renseignements claviers sont le verrouillage de certaines touches selon le modèle de clavier.

Dans la partie MULTIVUE, s'affichent les blocs de données des tâches interpréteur concernant cette console. Ces blocs de données se composent de la façon suivante :

- numéro de tâche pour cette console (de 1 à 4)
- attente de la réception d'un caractère ou attente de disponibilité d'écran pour le graphique symbolisé par *
- dernier numéro d'erreur système

Le bloc de données de la tâche interpréteur visualisée à l'écran est affiché en inversion vidéo.



7.	CONFIGURATION DE PROLOGUE	7-1
7.1.	Introduction	7-1
7.2.	Nomenclature des fichiers système	7-2
7.3.	Paramètres de configuration du noyau	7-6
7.3.1.	Description des paramètres communs	7-6
7.3.2.	Description des paramètres de poste	7-9
7.3.3.	Implantation physique des paramètres	7-12
7.3.4.	Le configurateur PROLOGUE	7-14
7.4.	Configuration des modules extension	7-15
7.4.1.	Configuration des modules d'impression	7-17
7.4.2.	Configuration des méthodes d'accès	7-19
7.4.3.	Configuration du disque mémoire	7-21
7.4.4.	Configuration du noyau de gestion graphique	7-22
7.4.5.	Configuration des utilitaires résidents	7-24
7.4.6.	Autres configurations	7-26
7.4.7.	Utilitaire de configuration : "SYSCONF-X"	7-27

7. CONFIGURATION DE PROLOGUE

7.1. INTRODUCTION

Pour pouvoir s'adapter aux conditions d'exploitation les plus variées, PROLOGUE définit un ensemble de paramètres de configuration que l'utilisateur peut modifier à son gré préalablement à l'utilisation de son système. Lorsque vous recevez un système PROLOGUE sur une ou plusieurs disquettes, il est configuré d'une façon standard et minimum. Un utilitaire, le CONFIGURATEUR, dénommé CF xxxx (selon le modèle de votre micro-ordinateur) permet l'adaptation de votre système. Cet outil de configuration, simple à mettre en œuvre, est décrit dans votre Guide d'Installation de PROLOGUE. La façon de configurer décrite dans ce chapitre doit être exceptionnelle. Ce chapitre vous décrit comment configurer le système PROLOGUE : Description des paramètres de configuration, mise à jour de ces paramètres directement dans le fichier SYSTEME à l'aide de l'utilitaire PATCH, description du fichier des extensions SYSCONF-S, mise en œuvre de l'utilitaire de configuration.

Les paramètres de configuration se décomposent en paramètres liés à PROLOGUE lui-même, appelés paramètres SYSTEME et les paramètres liés à la configuration matérielle utilisée, appelés paramètres ENVIRONNEMENT. Par exemple, le nombre de postes activés par PROLOGUE est un paramètre système, la voie V24 utilisée pour la procédure PTTY est un paramètre ENVIRONNEMENT. Les paramètres ENVIRONNEMENT étant spécifiques au matériel utilisé, ne sont pas décrits dans cette annexe. Il faut vous reporter dans ce cas au manuel de mise en œuvre et de configuration fourni avec le matériel.

7.2. NOMENCLATURE DES FICHIERS SYSTEME

SYSTEME

C'est le système PROLOGUE. Outre le noyau système proprement dit, invariant sur tout micro-ordinateur, ce fichier contient le noyau environnement, c'est à dire les contrôleurs périphériques de base. Le noyau environnement est spécifique à un type de matériel donné. Il permet l'adaptation du noyau système à ce matériel. Le contenu du noyau environnement est variable suivant les machines mais il contient le plus souvent les éléments suivants :

- Initialisation du matériel,
- Contrôleur unités de disquettes,
- Contrôleur unité de disque fixe,
- Contrôleur console du poste 0.

Au chargement du système, le fichier SYSTEME est chargé en mémoire par le chargeur de la machine, puis initialisé. Pour pouvoir être accessible par le chargeur, il est nécessaire que ce fichier respecte les conditions suivantes :

- Emplacement fixe et défini sur la disquette ou sur le disque.
- Disposition contigüe du fichier sur les secteurs disque.
- Absence de secteurs invalides à l'intérieur du fichier.

Le catalogue d'une disquette contenant le fichier SYSTEME apparaît comme suit :

VOL:PROLOGUE Nbfic:63/7 Lg.Gr:16 Occup:79/9

SYSTEME : 16 : (16,23)

/-X : 5 : (24,28)

CX : 4 : (32,35)

.....

.....

Vous remarquerez que le fichier système est situé à partir du granule 16. Il est constitué d'un seul bloc, ce qui garantit qu'il est écrit sur des secteurs consécutifs. Lorsque vous créez un volume PROLOGUE et que vous souhaitez y installer le fichier SYSTEME, il est recommandé après la création du volume, de copier ce fichier en premier. Il occupera naturellement une position correcte sur la disquette si la longueur des granules est de 16 secteurs.

Il est possible, si le micro-ordinateur le permet, de charger le fichier SYSTEME directement à partir du disque. Dans ce cas, il doit être copié sur le disque en respectant les indications données ci-dessous. Si la taille des granules du volume disque est de 16 secteurs, la recopie du fichier SYSTEME en premier après la création du volume le positionnera naturellement à la place adéquate.

SYSCONF-S

Fichier source éditable contenant la description des modules système extension devant être chargés en mémoire. Ces modules extension correspondent à des fonctionnalités optionnelles de PROLOGUE que l'utilisateur peut intégrer dans sa configuration. SYSCONF-S peut être édité sur l'écran ou sur l'imprimante. Le contenu de ce fichier est analysé par un programme utilitaire, SYSCONF-X, que l'interpréteur de commandes du poste 0 exécute systématiquement après chargement du système. Le fichier SYSCONF-S est recherché sur le disque implicite système (IS), défini par configuration. La structure du fichier SYSCONF-S est décrite de manière détaillée ci-dessous.

Il est nécessaire que tous les modules extension référencés dans SYSCONF-S soient présents sur la ressource disque implicite système (IS). Si un module extension est absent, un message le signalant est affiché, mais la configuration système se poursuit. Dans ce cas, il est de la responsabilité de l'utilisateur de déterminer si le système PROLOGUE est utilisable ou non.

SYSCONF—X

Programme utilitaire chargé d'analyser et de traiter le fichier SYSCONF-S. Il analyse la déclaration des modules extension, les charge en mémoire, et les initialise conformément aux paramètres d'initialisation définis. SYSCONF-X est recherché sur le disque implicite système (IS). SYSCONF-X et SYSCONF-S sont des fichiers optionnels de la configuration. Si l'un ou l'autre est omis, le système est réduit au noyau. Dans la plupart des cas, leur présence est cependant nécessaire.

YSERxx-S

Fichier source contenant les libellés des codes erreur système. Ce fichier est variable suivant la langue définie pour le système. Si le code langue est français, le nom du fichier est YSERF-S. Si le code langue est anglais ou autre, le nom du fichier est YSERGB-S. Le contenu du fichier peut être édité sur écran ou imprimante et modifié au gré de l'utilisateur en respectant la syntaxe suivante :

1 Ligne source par libellé. En tête de ligne, le numéro d'erreur doit être indiqué en décimal. Le libellé suit, séparé par moins 1 caractère blanc. Il est délimité par une cote "", de part et d'autre. Si la cote est employée dans le libellé lui-même, il faut la doubler.

Exemple :

```
01 'Unité Non Prête'  
47 'Clés d'Accès Invalides'
```

Si l'erreur survient en exploitation, son affichage est le suivant :

```
Erreur périphérique n° 1 Unité Non Prête  
Erreur Fichier n° 47 Clés d'accès Invalides
```

Le fichier YSERxx-S est optionnel. Si il est absent, un message le signale, mais l'initialisation du système se poursuit sans problème. Dans ce cas, il n'y a pas de libellé associé au code d'erreur.

Le fichier YSERxxx-S est recherché sur l'unité de disque implicite système, définie par configuration.

SYSTnn-C

Fichier source contenant les commandes initiales associées au poste nn. Ce fichier est traité par l'utilitaire ASG (voir chapitre 4). Il doit donc être conforme aux fichiers de type ASG. Ce fichier n'est pris en compte que si le code "commandes initiales" a été positionné pour le poste correspondant (voir ci-dessous). Pour le poste 0, le nom du fichier est SYST00-C. Pour le poste 7, il est SYST07-C. L'activation de ce fichier est équivalente à la commande

"-> ASG,SYSTnn-C<rc>".

Il est nécessaire de disposer de l'utilitaire ASG-X, présent sur le disque implicite programme du poste. Le fichier SYSTnn-C est recherché sur le disque implicite programme (IP) du poste.

ASG-X

Programme utilitaire nécessaire pour exécuter les commandes initiales de poste.

Il est nécessaire que tous les programmes et fichiers désignés dans les commandes initiales soient disponibles sur les unités de disque, de la même façon que pour des commandes utilisateur frappées au clavier.

7.3. PARAMETRES DE CONFIGURATION DU NOYAU

Les paramètres de configuration d'un système PROLOGUE se répartissent en trois catégories :

- Les paramètres du noyau système constants pour toute installation du système PROLOGUE sur n'importe quel micro.
- Les paramètres du noyau environnement liés à un matériel donné.
- Les paramètres associés aux modules extension définis dans SYSCONF-S.

Les paramètres du noyau environnement ne peuvent être décrits dans ce manuel, mais le sont dans le manuel d'installation et de mise en œuvre de PROLOGUE sur un matériel particulier.

Les paramètres de configuration des modules extension sont décrits de manière détaillée dans la documentation relative au module d'extension utilisé (voir bibliographie). Une récapitulation de ces paramètres est faite pour les modules extension les plus courants du système.

Les paramètres de configuration du noyau système se divisent en paramètres communs, généraux pour l'ensemble du système et paramètres de poste, spécifiques à chaque poste utilisateur.

7.3.1. Description des paramètres communs

KLAN Code Langue du Système en 10h

Définit la langue de base des messages du noyau système et des utilitaires. Le séparateur décimal est déduit du code langue, la règle étant que tous les pays européens, hormis la Grande-Bretagne, utilisent la virgule comme séparateur.

Les valeurs possibles sont les suivantes :

Code	Langue	Séparateur associé
0	Français	, (virgule)
1	Anglais	, (point)
2	Allemand	, (virgule)
3	Espagnol	, "
4	Italien	, "
5	Norvégien	, "
6	Suédois	, "
7	Danois	, "
8	Hollandais	, "
9	Finlandais	, "
10	Portugais	, "
11	Grec	, "

La valeur de KLAN est codée en binaire sur 1 octet.

Le noyau système, ainsi que les utilitaires de base son prévus pour être bilingues dans leur dialogue. Si le code langue est différent de 0, le dialogue sera anglais quel que soit le code langue effectif. Le code langue est accessible aux programmes B.A.L. ou assembleur qui peuvent l'utiliser à leur gré.

PMSYS Taille partition système en 11h-12h

Définit en nombre de Koctets la taille de la partition mémoire affectée au système. Ce paramètre doit être de 4 K par poste si la base de données (BD) est utilisée. La partition sert aux requêtes mémoire de petite taille (inférieure à 256 octets) ce qui limite l'émission de la mémoire. Si la taille indiquée est nulle, la partition système est inexistante et les requêtes sont automatiquement redirigées dans la partition commune. La valeur courante de ce paramètre est de 4 K x nombre de postes. La valeur est codée en binaire, en notation inversée sur 2 octets.

DSIS Ressource disque implicite système en 13, 14h, 15h

Définit la ressource disque implicite système commune à l'ensemble des postes. C'est sur cette ressource que seront recherchés les modules extension déclarés dans SYSCONF. La valeur de ce paramètre est donnée sous forme de 3 caractères ASCII indiquant le mnémonique et le numéro d'unité du disque implicite système commun. Exemple : "MD0".

NPOS Nb de postes actifs du système en 16h

Ce paramètre donne le nombre de postes actifs de la configuration. Il y a autant de tâches Interpréteur de commandes créées et activées à l'initialisation que de postes. La valeur de ce paramètre va de 1 pour une configuration monoposte à 8 qui est le nombre maximum de postes supportés par PROLOGUE 2.3. La valeur est donnée en binaire sur 1 octet.

CARIM1 Caractéristiques première imprimante en 1Bh-1Ch

Ce paramètre indique le nombre de lignes par page et le nombre de colonnes par ligne de la première imprimante déclarée du système. Ce paramètre peut être consulté par les programmes pour qu'ils s'adaptent aux caractéristiques de l'imprimante. Cependant, certains programmes ont leurs propres critères d'édition et ne tiennent pas compte de ce paramètre. La valeur du nombre de lignes est donnée en binaire sur 1 octet. La valeur du nombre de colonnes est donnée en binaire sur l'octet suivant.

7.3.2. Description des paramètres de poste

Chaque poste possède ses propres paramètres, dont les valeurs sont indépendantes de celles d'un autre poste. Elles sont situées de 80 h à 8Fh pour le premier poste.

DSIP Disque implicite programme (IP)

Ce paramètre indique la ressource disque implicite PROGRAMME du poste, c'est-à-dire l'unité de disque sur laquelle sont recherchés les programmes binaires désignés dans les commandes d'appel de PROLOGUE. La valeur de ce paramètre est donnée sous forme de 3 caractères ASCII indiquant le mnémonique et le numéro d'unité de cette ressource. Exemple : "MD0". La valeur par défaut est "FL0".

DSIF Disque implicite fichier (IF)

Ce paramètre indique la ressource disque implicite FICHIER du poste, c'est-à-dire l'unité de disque sur laquelle sont recherchés les fichiers référencés dans les commandes d'appel de PROLOGUE ou dans les programmes exécutés. La valeur de ce paramètre est donnée sous forme de 3 caractères ASCII indiquant le mnémonique et le numéro d'unité de cette ressource. Exemple : "MD0".

La valeur par défaut est "FL0".

DSIMP Ressource implicite d'impression

Ce paramètre indique la ressource implicite imprimante sur laquelle seront imprimés les caractères adressés à l'imprimante. La commande "CP,DF,sysconf-s,LO" éditera le contenu du fichier SYSCONF-S sur cette ressource implicite. La valeur de ce paramètre est donnée sous forme de 3 caractères ASCII indiquant le mnémonique et le numéro d'unité de cette ressource. Exemple : "IM0".

La valeur par défaut est "IM0".

PMPOS Taille de la partition mémoire du poste

Ce paramètre donne en K octets la taille de la partition mémoire affectée au poste. Les programmes exécutés sur ce poste seront exécutés dans cette partition mémoire. Une partition mémoire poste garantit une meilleure indépendance de chacun des postes, mais la mémoire totale nécessaire est plus importante. Si la valeur indiquée est nulle, il n'y a pas de partition mémoire attachée au poste et dans ce cas toutes les requêtes d'allocation sont redirigées dans la partition commune.

La valeur de ce paramètre est donnée en binaire et en notation inversée. La valeur à fournir doit être telle que le programme de taille maximum devant être exécuté sur ce poste puisse être chargé dans cette partition, et aussi que la somme des tailles des partitions de poste et de la partition commune soit inférieure à la taille mémoire disponible par PROLOGUE, de façon à ce que la taille de la partition commune soit suffisante. La valeur par défaut est 0, c'est-à-dire que la partition de poste est inexistante. Cette configuration convient dans la plupart des cas. La mémoire disponible pour PROLOGUE est donc partagée entre une partition système d'environ 4 K par poste, regroupant les requêtes de petites tailles et une partition commune importante.

PCOMIN Indicateur de commandes initiales en 8Bh

Ce paramètre indique la prise en compte du fichier "SYSTnn-C" contenant les commandes initiales du poste. 2 valeurs sont possibles : 0 indiquant qu'il n'y a pas de commandes initiales et 1 indiquant qu'il faut prendre en compte le fichier "SYSTnn-C". Par défaut la valeur est 0. La valeur est donnée en binaire sur 1 octet.

NPREX Protections d'exploitation

Ce paramètre contient un ensemble d'indicateurs autorisant ou non certaines fonctionnalités dangereuses. Chaque indicateur peut prendre la valeur 0 ou 1, 0 étant l'interdiction et 1 l'autorisation de la fonctionnalité.

Numéro d'indicateur Fonctionnalité associée

- | | |
|---|---|
| 0 | Fonctions "CP,CV,---", "CP,RV,---", "CP,DV,---" et "CPS,---" total autorisées ou interdites. |
| 1 | Ecriture de secteurs en format libre par "TMS,---", "PATCH,---", "CPS,---" partiel autorisées ou interdites. |
| 2 | Instruction système du B.A.L. : "PEEK", "POKE", "I/O", etc ... autorisées ou interdites. |
| 3 | Usage du configurateur (décrit ci-dessous) autorisé ou interdit. |
| 4 | Format de volume spécifique à la version 16 bits autorisé ou interdit. Si la fonctionnalité est interdite, les fonctionnalités spécifiques à la version 16 bits de PROLOGUE sont interdites, ce qui garantit une compatibilité 8 bits/16 bits des supports gérés. |
| 5 | Usage du traducteur B.A.L. et de l'option DEBUG de l'exécuteur autorisé ou interdit. |
| 6 | Format A1 autorisé (+ de 18 blocs) |

La valeur est indicateurs est donnée sur 2 octets. A chaque indicateur correspond un bit qui est à 0 ou 1 suivant la configuration. Le bit zéro, c'est-à-dire le plus à droite, correspond à l'indicateur 0. Par défaut tous les indicateurs sont positionnés à 1.

7.3.3. Implantation physique des paramètres

Les paramètres de configuration du noyau système sont définis à des adresses fixes dans le fichier binaire correspondant. L'adresse des paramètres est donnée sous forme d'un offset par rapport au début du binaire. Dans le fichier, le binaire commence effectivement au secteur 2. Les secteurs 0 et 1 étant l'en-tête du fichier. Dans le fichier SYSTEME, qui est présent sur la disquette ou le disque système, le binaire du noyau système est à la suite du binaire du noyau environnement. La position des paramètres est donc variable. Il est cependant assez facile de les localiser à l'aide de l'utilitaire PATCH et de la commande recherche de chaîne.

L'ensemble des paramètres système est contenu dans une page du binaire (256 octets) appelée page de configuration. La structure de cette page est la suivante :

OFFSET

00h/02h	Instruction JMP pour traitements des appels faits par le module d'initialisation environnement.
03h/05h	Relais avant exécution d'un programme binaire chargé par l'interpréteur de commandes. Ce relais fixe permet la pose d'un piège pour une exécution sous contrôle du moniteur de mise au point MM.
06h/07h	Octets réservés à zéro binaire.
08h	Indicateur de structure de la page de configuration. Valeur fixe = 31 h.
09h/0Fh	Octets réservés à zéro binaire.
10h	Code langue du système.
11h/12h	Partition système, donnée en binaire et en notation inversée.
13h/15h	Ressource disque implicite système. 3 caractères ASCII.
16h	Nb de postes actifs. 1 octet binaire de 1 à 8.

17h/1Ah	Octets réservés à zéro binaire.
1Bh/1Ch	Caractéristiques de la première imprimante. Le premier octet donne le nombre de lignes/page, le deuxième donne le nombre de colonnes/ligne.
1Dh/1Fh	Octets réservés à zéro binaire.
20h/3Fh	Zone réservée pour la sérialisation de PROLOGUE.
40h/7Fh	Octets réservés à zéro binaire.
80h/8Fh	Paramètres du poste 0.
80h/82h	Ressource disque implicite programme. 3 caractères ASCII.
83h/85h	Ressource disque implicite fichier. 3 caractères ASCII.
86h/88h	Ressource implicite d'impression. 3 caractères ASCII.
89h/8Ah	Partition mémoire du poste en Koctets.
8Bh	Indicateur de commandes initiales.
8Ch/8Dh	Protections d'exploitation.
8Eh/8Fh	Octets réservés à zéro binaire.
90h/9Fh	Paramètres du poste 1.
.....	
F0h/FFh	Paramètres du poste 7.

Les paramètres de configuration système sont donc contenus dans le fichier SYSTEME lui-même. Ils peuvent être consultés et mis à jour à l'aide de l'utilitaire PATCH. Cependant cette méthode demande une bonne connaissance du système et beaucoup de soins. Les modifications faites par PATCH sont immédiatement mises à jour dans le binaire. La nouvelle valeur des paramètres ne devient effective qu'après un chargement du système en mémoire.

7.3.4. Le configurateur PROLOGUE

Un programme de configuration auto-documenté est fourni avec PROLOGUE. Ce programme dont le nom est "CFxxxx" permet la mise à jour contrôlée de la plupart des paramètres décrits ci-dessus. Dans la plupart des cas, le configurateur sera suffisant pour configurer un système.

Ce programme permet une configuration complète du noyau système.

Ce programme de configuration est expliqué en détail dans votre Guide d'Installation de PROLOGUE.

La configuration de SYSCONF peut aussi être faite à l'aide d'un éditeur de textes, EDV par exemple, suivant les indications fournies au paragraphe 7.4.

L'activation du configurateur se fait de la façon suivante :

—> CFxxxx <rc>

ou

—> EX,CFxxxx <rc>

C'est un programme B.A.L. dont le suffixe est "T". Au programme de configuration est associé un fichier de messages "CFxxLA—S" (où LA représente la langue des messages) obligatoire pour son exécution et devant être sur l'unité Implicite Fichier. La mise à jour des paramètres se fait par une succession de menus interactifs et auto-documentés. La configuration s'achève par une mise à jour du fichier SYSTEME sur l'unité désignée, et la création et mise à jour des fichiers SYSCONF—S et, éventuellement, SYSTxx—C.

7.4.

CONFIGURATION DES MODULES EXTENSION

Chaque module extension est déclaré dans un fichier source éditable, "SYSCONF-S", de la manière suivante :

— MODULE (paramètres d'init) < LIEN (paramètres de lien)

"MODULE" correspond au fichier "MODULE-X" qui doit être présent sur l'unité de disque implicite système.

Les paramètres d'initialisation sont transmis successivement au module après son chargement en mémoire et en permettent une configuration dynamique. "LIEN" correspond soit à un module précédemment chargé auquel on désire raccrocher ce module, soit à un mot-clé correspondant à un module virtuel pré-intégré dans PROLOGUE. Les paramètres de lien permettent d'agir sur la nature du lien du module.

Les mots-clés prédéfinis par PROLOGUE 2.3. sont les suivants :

LO mot-clé indiquant que le module extension qui lui est rattaché traite les appels implicites à l'imprimante. Les rappels sont de type caractère. Par exemple la commande " $\rightarrow$ /,lis=lo" adressera directement le module d'extension rattaché à LO, caractère à caractère.

Ce mot-clé ne définit pas de paramètres de lien.

CO mot-clé indiquant que le module extension qui lui est rattaché traite les requêtes concernant le contrôleur console : affichage caractère, lecture caractère au clavier, etc ... Ce mot-clé ne définit pas de paramètres de lien.

ES mot-clé indiquant que le module extension qui lui est rattaché est "vu" comme un contrôleur périphérique accédé par des fonctions de niveau E/S. Ce mot-clé définit les paramètres de lien suivants :

p1 = mnémonique ressource. Exemple : p1 = 'DS'

p2 = nombre d'unités. Exemple p2 = 2

p3 = nature de la ressource gérée par le contrôleur.

Soit ressource disque, auquel cas p3 = 'NGF'.

Soit ressource spéciale, auquel cas p3 = 'méthode'

'méthode' désigne le nom de la méthode d'accès associée à cette ressource.

NGF mot-clé indiquant que le module extension qui lui est rattaché est une méthode d'accès "disque" reposant sur le NGF. Ce mot clé définit les paramètres de lien suivants :

p1 = numéro du relais système utilisé pour accéder à cette méthode d'accès.

En général, p1 = 32 ou 33 (= 20 h ou 21 h).

p2 = niveau hiérarchique de la méthode d'accès. Ce niveau peut prendre 4 valeurs possibles codées de 0 à 3.

SFX mot-clé indiquant que le module extension qui lui est rattaché est une méthode d'accès spéciale "hors disque" ne reposant pas sur le NGF. SFX définit deux paramètres de lien, identiques à ceux de NGF.

INTR mot-clé indiquant que le module extension qui lui est attaché est un programme binaire de type 'X' qui sera installé en résident par PROLOGUE à l'initialisation du système.

Ce mot-clé ne définit pas de paramètre de lien.

Les paramètres d'init ou de lien sont optionnels. De même la présence du champ lien est optionnelle. Un module peut également être lié plusieurs fois. Chaque module extension définit sa propre syntaxe de définition. Nous allons donner ci-dessous les déclarations les plus courantes que vous aurez à faire.

NOTE :

Le caractère ";", en première colonne permet d'introduire des commentaires dans le fichier SYSCONF-S. Tous les caractères qui le suivent jusqu'à la fin de ligne sont ignorés. De même, les caractères "" (blanc) et TAB (tabulation) sont ignorés. Les exemples ci-dessous peuvent donc être recopiés textuellement dans le fichier SYSCONF-S.

Les modules extension se partagent en modules extension SYSTEME, indépendants du matériel sur lesquels ils sont utilisés, et les modules extension environnement, spécifiques à un matériel particulier. Le nom des modules environnement est noté par "MODxx", 'xx' indiquant que le nom du module varie suivant le matériel.

La déclaration des modules environnement est faite ici à titre indicatif, sans garantie absolue d'exacte conformité.

7.4.1. Configuration des modules d'impression

Les modules d'impression sont au nombre de 2 ;

"SPLO1-X" assure la gestion de 1 à 4 imprimantes dans un contexte mono ou multiposte. SPL01 permet d'accéder à l'imprimante comme une ressource périphérique ayant un mnémonique et définissant sa propre méthode d'accès conforme à l'interface "SF" défini par PROLOGUE. Dans ce cas l'imprimante n'est plus accédée implicitement caractère à caractère, mais par des ordres d'ouverture, d'écriture et de fermeture. SPLO1 permet aussi de travailler en mode "SPOUL", avec une gestion de listes à éditer sur disque, sous forme de fichiers temporaires d'impression. Le mode SPOUL est mis en œuvre par l'utilitaire "SPOOL" décrit au chapitre 4.

"IMPRL-X" c'est le contrôleur physique permettant d'adapter PROLOGUE à une imprimante particulière. Ce module appelle un fichier paramètre spécifique à l'imprimante concernée.

La déclaration de ces modules dans SYSCONF-S se fait de la façon suivante :

; Gestion complète Imprimante
; -----

– SPLO1 < SFX (p1=32,p2=0) ; rattachement de SPLO1 en tant que
; méthode d'accès hors NGF

< ES (p1='IM',p2=2,p3='SPLO1') ; affectation d'un mnémonique ("IM")
; d'un nombre d'unités (P2=2) à la
; ressource impression gérée par
; SPLO1

< LO

; rattachement de SPLO1 au mot-clé
; "LO" de façon à ce qu'il récupère
; les appels implicites à l'imprimante

— IMPRL (P1=LINA11) < SPLO1

; rattachement du contrôleur impri-
; mante à SPLO1
; cette imprimante sera vue comme
; "IMO"

Si la déclaration de ces modules est faite de façon correcte, le mnémonique de l'unité implicite d'impression du contexte de poste doit apparaître. Par exemple :

—> Prologue Version 2.3. Poste 0 Implicites : FL0,FL1,IMO

Il est possible de simplifier la déclaration de la façon suivante :

; Déclaration simplifiée Imprimante

; -----

— IMPRL (P1=param) < LO ; rattachement direct du contrôleur imprimante
; au mot-clé "LO"

Cette déclaration implique les restrictions suivantes :

- Utilisation strictement monoposte.
- Fonction "SPOOL" non supportée.
- 1 seule imprimante.
- L'imprimante ne peut être adressée qu'implicitement.
- Il n'y a pas de mnémonique attaché à cette ressource.

Dans ce cas, le mnémonique de l'unité implicite d'impression n'apparaît plus dans le contexte de poste. Par exemple :

—> Prologue Version 2.3. Poste 0 Implicites : FL0,FL1,---

7.4.2. Configuration des méthodes d'accès

RAPPEL :

PROLOGUE supporte les méthodes d'accès suivantes :

- Accès NGF : Noyau de Gestion de Fichiers
- Accès SEQ : Séquentiel pur
- Accès SI : Séquentiel Indexé
- Accès MC : Multicritère
- Accès BD : Base de données

Les méthodes d'accès NGF et SEQ font partie du noyau système et sont donc toujours disponibles dans une configuration PROLOGUE. Les méthodes d'accès SI, MC et BD sont gérées par des modules optionnels indépendants qu'il est nécessaire de déclarer dans SYSCONF.

- Le module "SI-X" correspond à l'accès Séquentiel Indexé.
- Le module "MC-X" correspond à l'accès Multicritère et impose la présence du module SI.
- Le module "BD-X" correspond à l'accès Base de Données et impose la présence du module MC et donc de S.I.

La déclaration de des modules dans SYSCONF se fait de la façon suivante :

; Déclaration de l'accès Séquentiel Indexé

; -----

- SI < NGF (p1=32,p2=2) ; rattachement du SI au mot-clé NGF. C'est une
; méthode d'accès "disque". Les paramètres
; p1 et p2 définissent le niveau d'accès du module.

; Déclaration de l'accès Multi-critères

; -----

— MC (p1=n) < NGF (p1=32,p2=3)

; rattachement du MC au mot-clé
; NGF. C'est une méthode d'accès
; "disque". Le paramètre "p1=n"
; définit le nombre maximum de fi-
; chiers MC que l'on peut ouvrir
; simultanément. Les valeurs possi-
; bles pour n sont comprises entre 1
; et 31 pour l'ensemble du système.
; Ce paramètre peut être omis. Sa
; valeur par défaut est 4.

; Déclaration de l'accès Base de Données

; -----

— BD < NGF (p1=33,p2=2)

; rattachement du module BD au mot
; clé NGF. C'est une méthode d'accès
; "disque".

< NGF (p1=33,p2=3)

; BD possède 2 niveaux d'accès.

IMPORTANT :

La méthode d'accès, Base de Données (BD) ne fait pas partie de la distribution du système PROLOGUE de base. Les indications données ci-dessus supposent donc que vous avez fait l'acquisition de cette option lors de la commande de votre système.

Si vous ne l'avez pas fait, les modules binaires correspondant à cette méthode d'accès ne sont pas inclus dans les disquettes de distribution que vous aurez reçues. Vous avez la possibilité d'en faire l'acquisition séparément.

7.4.3. Configuration du disque mémoire

Le module extension disque mémoire permet la simulation d'une unité de disque en mémoire vive. Il définit donc une nouvelle ressource disque ayant son mnémonique propre et accédée d'une façon identique à une ressource disque. L'intérêt du disque mémoire est d'offrir un accès très rapide à des fichiers sollicités fréquemment. En contrepartie, il nécessite une plus grande taille mémoire. De plus, tous les fichiers créés sur cette ressource sont perdus lors de l'arrêt du système. En conséquence, il faut prévoir une initialisation systématique de cette ressource lors du chargement système, et sauvegarder toutes les données importantes qui s'y trouvent avant arrêt du système.

Le disque mémoire définit un disque virtuel dont les caractéristiques sont les suivantes :

- mnémonique = 'RM'
- nombre d'unités = 1
- nombre de secteurs/piste = 16
- taille d'une piste : 4 K octets
- nombre total de pistes = défini par configuration

Le nom de ce module extension est "MEMDISK-X". Sa déclaration dans SYSCONF se fait de la façon suivante :

```
; Déclaration disque mémoire  
; -----
```

```
— MEMDISK (p1=n) < ES (p1='RM',p2=1,p3='NGF')
```

Avec n = nombre de 'pistes' mémoire de la ressource
= taille de la ressource en unités de 4 K octets.

Cette ressource disque mémoire sera adressée par "RM0". Vous pourrez y pratiquer par exemple les opérations suivantes :

Création d'un volume :

```
—> CP,CV,rm0,MEMOIRE<rc>
```

Recopie des fichiers binaires de la disquette :

```
—> CP,DF,fl10.*-X,rm0<rc>
```

etc ...

Ce module est totalement portable d'un matériel à l'autre.

Il est portable.

; Déclaration du Noyau de Gestion Graphique

– SFGR < SFX (p1=32,p2=0) ; rattachement de SFGR au mot-clé SFX.
SFGR est donc vu comme une méthode d'accès hors NGF.

- ESGR < ES (p1='EC',p2=1 ,p3='SFGR') ; rattachement du module ESGR
- ; en tant que contrôleur périphé-
- ; que de niveau E/S. Le périphé-
- ; rique graphique est désigné par
- ; "EC0".

- $\text{COGR}_{xx} < \text{ESGR}$; rattachement du contrôleur
; physique à ESGR

Remarque :

La description faite ci-dessus représente la configuration la plus courante du Noyau de gestion graphique. Des informations plus détaillées peuvent être obtenues dans la documentation "Manuel de référence B.A.L. Graphique".

IMPORTANT :

Le noyau de Gestion graphique ne fait pas partie de la distribution du système PROLOGUE en version de base. Les indications données ci-dessus supposent donc que vous avez fait l'acquisition de ce produit lors de la commande de votre système PROLOGUE. Si ce n'est pas le cas, les modules binaires correspondants ne sont pas inclus dans les disquettes de distribution que vous avez reçues. Vous avez la possibilité d'en faire l'acquisition séparément.

; Déclaration d'imprimante graphique

; -----

— SGLOG < SFX (P1=32,P2=0)

< ES (P1=IG,P2=1,P3='SGLOG')

7.4.5. Configuration des utilitaires résidents

PROLOGUE 2.3 donne la possibilité d'installer en mémoire certains utilitaires de base, parmi les plus courants. L'installation d'un utilitaire en résident se fait par une déclaration particulière de cet utilitaire dans le configurateur ou le fichier SYSCONF-S.

Lors de l'initialisation système, chaque utilitaire déclaré ainsi est chargé en mémoire une fois pour toutes. Lors de l'exploitation du système, toute référence faite à un utilitaire résident exécute directement cet utilitaire sans procéder à son chargement. De plus certains utilitaires permettent le partage de leur code entre plusieurs utilisateurs.

Dans la description faite de chaque utilitaire dans le chapitre 4, il est indiqué si l'utilitaire peut être installé en résident et quelle est la déclaration correspondante pour l'installer en mémoire.

L'installation d'un utilitaire en résident en mémoire se fait par la déclaration suivante dans le fichier SYSCONF :

– Utilitaire < INTR

où

'utilitaire' est le nom de l'utilitaire, c'est-à-dire le nom du fichier binaire correspondant, de suffixe 'X'.

'INTR' est un mot-clé signalant à l'interpréteur de commandes de PROLOGUE la présence de cet utilitaire en résident.

Cette déclaration ne peut s'appliquer qu'à des utilitaires autorisant la fonctionnalité.

Le tableau ci-après est une récapitulation des possibilités qu'ont chacun des utilitaires de base d'être installés en mémoire et de permettre le partage de leur code.

UTILITAIRE	RESIDENT	PARTAGEABLE	NB MAX DE PARTAGES
/-X (catalogue	OUI	OUI	illimité
CP-X	OUI	OUI	illimité
DATE-X	OUI	NON	-----
VPMI-X	OUI	OUI	illimité
CPS-X	NON	----	-----
SV-X	OUI	NON	-----
RT-X	OUI	NON	-----
ASG-X	OUI	NON	-----
SPOOL-X	NON	----	-----
EDV-X	OUI	OUI	illimité
PATCH-X	NON	----	-----
STATUS-X	OUI	NON	-----
STMEM-X	OUI	NON	-----
SVSI-X	OUI	NON	-----
RTSI-X	OUI	NON	-----
EX-X	OUI	NON	-----

Programmes optionnels

TR-X	OUI	OUI	illimité
TELE-X	OUI	NON	-----

IMPORTANT

Les programmes optionnels nommés ci-dessus ne font pas partie de la distribution de PROLOGUE en version minimum. Ils peuvent donc être absents des disquettes de distribution que vous avez reçues.

7.4.6. Autres configurations

Les paragraphes précédents décrivent la façon de configurer le fichier SYSCONF.S pour intégrer les modules extensions les plus courants utilisés par PROLOGUE.

Il existe cependant de nombreuses autres possibilités de configuration correspondant soit à des fonctionnalités optionnelles de PROLOGUE et non décrites ici, soit à des modules extensions spécifiques à un environnement matériel particulier.

Pour chacun des produits extension disponibles sous PROLOGUE il existe une documentation de référence décrivant notamment de manière détaillée la façon de déclarer les modules extension correspondants dans SYSCONF.

Exemple :

La façon de configurer les modules extensions composant le produit de télécommunication "X25" est décrite dans la documentation associée :

"Procédure X25 : Guide de l'utilisateur"

La façon de configurer les modules extension spécifiques à un environnement particulier est décrite dans la documentation spécifique à un matériel donné et qui décrit tous les aspects particuliers de l'implantation de PROLOGUE sur ce matériel.

Ce manuel étant relatif à l'utilisation de PROLOGUE dans son ensemble n'aborde jamais d'aspects qui soient spécifiques à une machine en particulier. C'est pourquoi le nom des modules extensions environnement est donné sous la forme "MODxx", 'xx' représentant un sigle spécifique du matériel utilisé.

Note :

Voir cependant le chapitre 6 pour configurer les extensions de PROLOGUE.

7.4.7. L'utilitaire "SYSCONF-X"

Le programme utilitaire "SYSCONF-X" a pour rôle d'analyser le fichier de configuration SYSCONF-S. Il assure les fonctions suivantes :

- analyse syntaxique des déclarations,
- recherche et chargement des modules en mémoire,
- initialisation du module avec passation des paramètres fournis dans la déclaration,
- localisation du module ou mot-clé "lien",
- lien du module avec passation des paramètres de lien.

L'exécution de ce programme est automatique après le chargement du noyau système. La commande "`-> IS.SYSCONF<rc>`" est préconstruite dans le système. Elle est exécutée par l'interpréteur de commandes du poste 0. Si la configuration du système est multiposte, l'interpréteur de commandes des postes extension est bloquée pendant le temps d'exécution de SYSCONF. Lorsque son exécution est terminée, le symbole "`-`" est affiché à la suite de la commande et le signal d'appel de PROLOGUE, "`->`" est affiché sur le poste 0 et les postes d'extension sont débloqués. L'affichage sur l'écran du poste 0 est le suivant :

```
PROLOGUE Version 2.3.  
n° de série : xxxxxxxxxxxx  
Copyright (c) 1984 BULL Micral  
  
-> IS.SYSCONF -  
-> -
```

La notation "IS." spécifie que SYSCONF-X sera recherché sur le disque implicite système de PROLOGUE. De même le fichier SYSCONF-S ainsi que les modules extension référencés dans ce fichier seront également recherchés sur le disque implicite système.

Dans un usage normal de PROLOGUE, l'exécution de SYSCONF-X est automatique, et l'utilisateur n'a pas à se soucier de la syntaxe d'appel de cet utilitaire. Si vous tentez de réexécuter SYSCONF une deuxième fois, vous obtenez le message suivant :

```
-> IS.SYSCONF -  
-> SYSCONF<rc>  
    Configurateur Système Version 1.3a  
    Copyright (c) 1984 BULL Micral  
  
    SYSCONF ne peut être exécuté plusieurs fois  
  
-> _
```

Ceci vous permet d'obtenir la bannière d'identification de cet utilitaire.

Dans certains cas, il peut être intéressant de différer l'exécution de SYSCONF ou de l'exécuter avec des paramètres différents. Ceci est surtout intéressant dans une phase de mise au point.

Si vous souhaitez différer l'exécution de SYSCONF, il suffit de le renommer, par exemple en "SYSCONF1-X". Dans ce cas, l'initialisation du système sera la suivante :

```
"""" bannière PROLOGUE """"  
  
-> IS.SYSCONF  
    SYSCONF : Programme inexistant  
  
-> _
```

Vous avez alors la possibilité de lancer manuellement SYSCONF, de la même façon que vous exécutez une commande utilitaire.

La syntaxe générale d'appel de SYSCONF est la suivante :

-> [unité.] SYSCONF1[,NFIC=desfic] [,ADR] <rc>

avec :

unité ressource disque sur laquelle sera recherché le fichier SYSCONF-X. Par défaut c'est la ressource disque implicite programme du poste.

SYSCONF1 nom du fichier binaire, tel qu'il a été renommé, de suffixe "X" correspondant au programme SYSCONF

NFIC= mot-clé précisant la nature du paramètre qu suit

desfic fichier source contenant la déclaration des modules extension à installer en mémoire. Par défaut le nom de ce fichier source est SYSCONF-S.

Il est parfois intéressant d'exécuter SYSCONF à partir d'un fichier de configuration différent du fichier standard SYSCONF-S. Cela permet de définir des configurations provisoires sans affecter le fichier SYSCONF-S. Il n'y a pas de suffixe implicite attribué au fichier.

ADR mot-clé permettant l'affichage successif des modules chargés en mémoire ainsi que leur adresse physique d'implantation. Cette information n'a d'intérêt que pour un programmeur système.

Cette option est incompatible avec Multivue.

Remarque :

Lorsqu'un module est déclaré dans SYSCONF-S, sa déclaration ne fait en général pas mention explicitement à une unité de disque. Dans ce cas, c'est systématiquement la ressource disque implicite système de PROLOGUE qui est prise par défaut pour rechercher les modules extensions.

Cependant, il est possible de désigner explicitement une unité de disque particulière, suivant la syntaxe habituelle de PROLOGUE. Par exemple :

— FL1.SI < NGF (p1=32,p2=2)

chargera le module "SI-X" à partir de l'unité "FL1", quelle que soit l'unité implicite système. Cette possibilité peut être intéressante dans certains cas, par exemple si vous souhaitez utiliser une version temporaire de module, sans avoir à écraser la version courante.

Message d'erreurs de SYSCONF-X

Lorsque l'exécution de SYSCONF est correcte, aucun message ne doit apparaître et le symbole "—" est affiché à la suite de la ligne de commande :

—> IS.SYSCONF —
—> _

Outre les erreurs systèmes, SYSCONF émet quelques messages d'erreur propres. Le nom du module en cause est systématiquement affiché avec le libellé de l'erreur.

En général, si une erreur survient, son libellé est affiché et le traitement de la configuration système est abandonné. Dans ce cas, il est recommandé de supprimer l'erreur en recherchant la cause et de recharger le système. L'usage d'un système pour lequel le traitement de SYSCONF aurait provoqué une erreur fatale peut conduire à des résultats imprévisibles.

Seul le message "FICHIER INEXISTANT" est un avertissement qui ne provoque pas l'abandon de la configuration.

ERREUR SUR NOM DE MODULE

Le nom du module traité est incorrect.

Exemple : trop de caractères ou caractère interdit dans sa désignation.

ADRESSE DE CHARGEMENT INCORRECTE

Cette erreur est une erreur interne du système et ne doit en principe jamais survenir.

STRUCTURE DU BINAIRE INCORRECTE

Le contenu du fichier binaire correspondant au module est incorrect. Remettre à jour ce fichier à partir d'une référence.

ESPACE MEMOIRE LIBRE INSUFFISANT

L'occupation mémoire des modules à charger est supérieure à la taille mémoire disponible sur le matériel utilisé. Trois solutions sont possibles :

- Supprimer la déclaration de certains modules.
- Modifier les paramètres d'initialisation.
- Augmenter la taille mémoire physique.

CHARGEMENT REFUSE

Un incident s'est produit lors du chargement d'un module (par exemple : erreur de lecture).

FICHER INEXISTANT

Le fichier binaire correspondant au module ne figure pas sur l'unité de disque désignée dans la déclaration. Ce message est un avertissement et la configuration est poursuivie. Il appartient à l'utilisateur de déterminer si l'absence de ce module est volontaire ou ne provoque pas d'incohérences dans la configuration du système.

ERREUR DE SYNTAXE DANS LES PARAMETRES

La déclaration des paramètres n'est pas conforme à la syntaxe imposée par SYSCONF.

ERREUR DANS LE NUMERO D'UN PARAMETRE

Caractère illicite dans le numéro d'un paramètre ou valeur trop élevée.

ERREUR DANS LA VALEUR D'UN PARAMETRE

La valeur donnée d'un paramètre n'est pas conforme aux règles syntaxiques imposées par SYSCONF.

INITIALISATION INCORRECTE

Le module traité rejette l'initialisation qui lui est proposée. La cause peut être due à la fourniture de mauvais numéros de paramètres ou de valeurs non conformes à ce qu'il attend.

Un numéro d'erreur, retourné par le module est affiché.

CHAINAGE INCORRECT

Le lien du module traité vers un autre module ou un mot-clé est refusé. Le numéro de l'erreur retourné par le module lien est affiché.

ERREUR DE SYNTAXE DANS LE FICHIER "SYSCONF-X"

La séparation entre les paramètres d'initialisation et le nom du lien n'est pas faite par un caractère "<". Une zone paramètre ne commence pas par un "(" ou ne finit pas par un caractère ")".

LIBELLES DES CODES ERREURS NON CHARGES

Le fichier "SYSERxx-S" est absent de la ressource disque commune. Ce message est préventif et n'empêche pas PROLOGUE de fonctionner correctement.

Dans ce cas, si une erreur système survient en cours d'exploitation le libellé explicatif associé au numéro de l'erreur n'est pas affiché.

TABLE DE CODIFICATION DES CARACTERES

La table ci-après indique la codification interne des caractères, définie par PROLOGUE. A chaque caractère est associée une valeur entière codée de 0 à 255 (soit 00h à ffh en hexadécimal). Toutes les valeurs de la table ne correspondent pas forcément à des caractères affichables. Les caractères dont la valeur est comprise entre 0 et 31 (00h à 1fh), ainsi que le caractère 7fh sont des caractères de contrôle, en général non affichables. La codification des caractères de 0 à 127 (00h à 7fh) est conforme à la norme de codification ASCII. Les caractères codés de 128 à 255 (80h à ffh) représentent une extension du jeu standard ASCII et incorpore des configurations de caractères nationaux, ainsi que divers symboles d'encadrement ou mathématiques.

La codification définie par PROLOGUE répond aux besoins de la totalité des pays utilisant un alphabet LATIN. Les fichiers de textes créés par PROLOGUE sont portables d'un pays à l'autre.

ANNEXE A

A.1 CODE PROLOGUE

Y X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	ESPACE	0	@	P	`	p	Ç	É	á	DENSITÉ FAIBLE DE POINTS	☐	Ø	∞	≡
1	SOH	DC1	!	1	A	Q	a	q	ü	æ	í	MOYENNE	☐	¨	β	±
2	STX	DC2	"	2	B	R	b	r	é	Æ	ó	FORTE	☐	§	γ	≥
3	ETX	DC3	#	3	C	S	c	s	â	ô	ú	☐	☐	¶	π	≤
4	EOT	DC4	\$	4	D	T	d	t	ä	ö	ñ	☐	☐	'	Σ	f
5	ENQ	NAK	%	5	E	U	e	u	à	ò	Ñ	À	☐	☐	σ	f
6	ACK	SYN	&	6	F	V	f	v	â	û	à	Á	Ï	œ	μ	÷
7	BEL	ETB	'	7	G	W	g	w	ç	ù	ó	Â	Ò	Œ	τ	≈
8	BS	CAN	(	8	H	X	h	x	ê	ÿ	¿	Ã	Ó	←	Φ	°
9	HT	EM	)	9	I	Y	i	y	ë	Ö	▯	È	Ô	▯	⊖	•
A	LF	SUB	*	:	J	Z	j	z	è	Ü	▯	Ê	Õ	▯	Ω	•
B	VT	ESC	+	;	K	l	k	l	ï	ç	½	Ë	Ù	▯	δ	√
C	FF	FS	,	<	L	\	l	l	î	£	¼	Ì	Ú	▯	∞	η
D	CR	GS	-	=	M	l	m	l	ì	¥	ì	Í	Û	▯	∅	²
E	SO	RS	.	>	N	^	n	~	Ä	Pt	»	Î	ã	▯	∈	▯
F	SI	US	/	?	O	_	o	DEL	A	f	«	☐	õ	▯	∩	ESPACE

AVEC YX EN HEXADÉCIMAL

Exemple : "Z" Code 5A hexadécimal

ANNEXE. CODES CARACTERES ASCII

Equivalence Décimal, Hexadécimal, ASCII

<u>CAR</u>	<u>DEC</u>	<u>HEX</u>	<u>CAR</u>	<u>DEC</u>	<u>HEX</u>	<u>CAR</u>	<u>DEC</u>	<u>HEX</u>
NUL	000	00	!	033	21	B	066	42
SOH	001	01	"	034	22	C	067	43
STX	002	02	#	035	23	D	068	44
ETX	003	03	\$	036	24	E	069	45
EOT	004	04	%	037	25	F	070	46
ENQ	005	05	&	038	26	G	071	47
ACK	006	06	'	039	27	H	072	48
BEL	007	07	(	040	28	I	073	49
BS	008	08	)	041	29	J	074	4A
HT	009	09	*	042	2A	K	075	4B
LF	010	0A	+	043	2B	L	076	4C
VT	011	0B	,	044	2C	M	077	4D
FF	012	0C	—	045	2D	N	078	4E
CR	013	0D	.	046	2E	O	079	4F
SO	014	0E	/	047	2F	P	080	50
SI	015	0F	0	048	30	Q	081	51
DLE	016	10	1	049	31	R	082	52
DC1	017	11	2	050	32	S	083	53
DC2	018	12	3	051	33	T	084	54
DC3	019	13	4	052	34	U	085	55
DC4	020	14	5	053	35	V	086	56
NAK	021	15	6	054	36	W	087	57
SYN	022	16	7	055	37	X	088	58
ETB	023	17	8	056	38	Y	089	59
CAN	024	18	9	057	39	Z	090	5A
EM	025	19	:	058	3A	[	091	5B
SUB	026	1A	;	059	3B	\	092	5C
ESC	027	1B	<	060	3C	]	093	5D
FS	028	1C	=	061	3D	^	094	5E
GS	029	1D	>	062	3E	—	095	5F
RS	030	1E	?	063	3F	'	096	60
US	031	1F	@	064	40	a	097	61
Espace	032	20	A	065	41	b	098	62

CAR	DEC	HEX	CAR	DEC	HEX	CAR	DEC	HEX
c	099	63	ï	139	8B		178	B2
d	100	64	î	140	8C		179	B3
e	101	65	ì	141	8D		180	B4
f	102	66	À	142	8E	À	181	B5
g	103	67	Ä	143	8F	Á	182	B6
h	104	68	É	144	90	Â	183	B7
i	105	69	æ	145	91	Ã	184	B8
j	106	6A	Æ	146	92	È	185	B9
k	107	6B	ô	147	93	Ê	186	BA
l	108	6C	ö	148	94	E	187	BB
m	109	6D	ò	149	95	Ì	188	BC
n	110	6E	û	150	96	Í	189	BD
o	111	6F	ù	151	97	Î	190	BE
p	112	70	ÿ	152	98		191	BF
q	113	71	Ö	153	99		192	C0
r	114	72	Ü	154	9A		193	C1
s	115	73	ƒ	155	9B		194	C2
t	116	74	£	156	9C		195	C3
u	117	75	¥	157	9D		196	C4
v	118	76	Ƨ	158	9E		197	C5
w	119	77	Pt	159	9F	I, O	198	C6
x	120	78	f	160	A0	Ó	199	C7
y	121	79	á	161	A1	Ô	200	C8
z	122	7A	í	162	A2	Õ	201	C9
~	123	7B	ó	163	A3	Ö	202	CA
~	124	7C	ú	164	A4	Ù	203	CB
~	125	7D	ũ	165	A5	Ú	204	CC
~	126	7E	ñ	166	A6	Û	205	CD
~	127	7F	Ñ	167	A7	ä	206	CE
DEL	128	80	ä	168	A8	õ	207	CF
Ç	129	81	å	169	A9	ø	208	D0
ü	130	82	ç	170	AA	·	209	D1
é	131	83	┐	171	AB	§	210	D2
â	132	84	└	172	AC	¶	211	D3
ä	133	85	1/2	173	AD	·	212	D4
à	134	86	1/4	174	AE	·	213	D5
ª	135	87	ì	175	AF	æ	214	D6
ç	136	88	»	176	B0	OE	215	D7
ê	137	89	«	177	B1		216	D8
ë	138	8A					217	D9

<u>CAR</u>	<u>DEC</u>	<u>HEX</u>
	218	DA
	219	DB
	220	DC
	221	DD
	222	DE
	223	DF
α	224	E0
β	225	E1
γ	226	E2
π	227	E3
Σ	228	E4
σ	229	E5
μ	230	E6
τ	231	E7
Φ	232	E8
Θ	233	E9
Ω	234	EA
δ	235	EB
∞	236	EC
ϕ	237	ED
$\in$	238	EE
$\cap$	239	EF
$\equiv$	240	F0
$\pm$	241	F1
$\geq$	242	F2
$\leq$	243	F3
$\int$	244	F4
$\div$	245	F5
$\approx$	246	F6
$\circ$	247	F7
$\bullet$	248	F8
$\bullet$	249	F9
$\bullet$	250	FA
$\sqrt{\quad}$	251	FB
η	252	FC
2	253	FD
!	254	FE
Espace	255	FF

A.3 ABBREVIATIONS STANDARD

Abréviations standards pour les caractères ASCII de 0 à 31 (00 à 1F hexadécimal)

HEX Abrév. Signification

06	ACK	Acknowledge	Accusé de Réception
07	BELL	Bell	Sonnerie
08	BS	Backspace	Espace Arrière
18	CAN	Cancel	Annuler
0D	CR	Carriage Return	Retour Chariot
11	DC1	Direct Control 1	Contrôle Direct 1
12	DC2	Direct Control 2	Contrôle Direct 2
13	DC3	Direct Control 3	Contrôle Direct 3
14	DC4	Direct Control 4	Contrôle Direct 4
10	DLE	Data Link Escape	Sortie de Liaison
19	EM	End of Medium	Fin de Support
05	ENQ	Enquiry	Interrogation
04	EOT	End of Transmission	Fin de Transmission
1B	ESC	Escape	Sortie
17	ETB	End Transmission Block	Fin Bloc de Transmission
03	ETX	End Text	Fin Texte
0C	FF	Form Feed	Saut à la Page
1C	FS	Form Separator	Séparateur d'Etat
1D	GS	Group Separator	Séparateur de Groupe
09	HT	Horizontal Tab	Tabulation Horizontale
0A	LF	Line Feed	Saut à la Ligne
15	NAK	Negative Acknowledge	Accusé de Réception Négatif
00	NUL	Null	Nul
1E	RS	Record Separator	Séparateur d'Enregistrement
0F	SI	Shift In	En code
0E	SO	Shift Out	Hors Code

01	S0H	Start of Heading	Début d'En-tête
0Z	STX	Start Text	Début de Texte
1A	SUB	Substitute	Substitution
16	SYN	Synchronous Idle	Synchronisation Repos
1F	US	Unit Separator	Séparateur de Sous-enregistrement
0B	VT	Vertical Tab	Tabulation Verticale

ANNEXE B : LISTE DES CODES-ERREURS

Les codes-erreurs sont des informations du système envoyées à l'utilisateur pour indiquer une situation anormale. Ils ont une valeur comprise entre 0 et 255 inclus. La valeur 0 indique une exécution correcte et n'est donc jamais affichée ; elle est le plus souvent remplacée par l'affichage du "—", notamment avec les utilitaires.

On peut classer les codes-erreurs en catégories :

Numéro des codes-erreurs	Catégorie	Pour plus de précision, se reporter ...
1 à 15	Périphérique	à ce manuel ou à celui du constructeur
16 à 19	S.O.R.	au manuel "SPR" (Guide utilisateur)
20 à 25-27 28-29-56-58	Système	au manuel "PROLOGUE-86" (Guide du programmeur système)
30 à 39	Commandes	à ce manuel
40 à 69 200 à 209	Gestion de fichiers N.G.F.	à ce manuel ou au manuel "PROLOGUE-86" (Guide du programmeur système)
70 à 82	Séquentiel indexé	à ce manuel
83 à 96	Multicritère	à ce manuel
97 à 127	B.A.L.	au manuel "Langage B.A.L." (Manuel de référence)
170 à 189	Base de données	"Système de gestion de base de données relationnelle" (Manuel de référence)
190 à 196	Graphique	au manuel B.A.L. graphique (Manuel de référence)

Excepté lors de l'exécution de programmes B.A.L., les codes-erreurs sont affichés de la manière standard suivante :

'Catégorie' no 'Libellé'

où

'Catégorie'	représente les catégories énumérées ci-dessous.
numéro	est le numéro d'erreur compris entre 1 et 255.
'Libellé'	est un bref message, optionnel, commentant l'erreur.

Exemple :

```
-> /,FL0
    Catalogue du volume Version 3.4b
    Copyright (c) 1984 BULL Micral
    Erreur périphérique no 1 Unité Non Prête
-> _
```

Les libellés de codes-erreurs ne sont définis que pour un certain nombre d'erreurs, parmi les plus importantes. Ces libellés sont définis dans le fichier "SYSERF-S" (Français) ou "SYSERGB-S" (Anglais ou autre langue), chargé en mémoire à l'initialisation du système. Ils peuvent être modifiés, complétés et même traduits directement par l'utilisateur, à l'aide d'un simple éditeur de textes.

Le mécanisme des codes-erreurs est également décrit dans le paragraphe 3.2.5.

Les codes-erreurs ci-après représentent tous les codes-erreurs de PROLOGUE. Certains ne sont pas liés directement au système d'exploitation, mais à certaines configurations. Toutefois, il a paru souhaitable de les avoir tous ici représentés afin que l'utilisateur n'ait qu'une seule source à consulter lorsqu'un incident se produit. Une fois l'erreur analysée, il pourra toujours se reporter au manuel de référence approprié, pour avoir plus de précisions sur la nature de l'incident ou les actions à entreprendre pour y remédier.

Numéro du code-erreur	Libellé associé	Commentaires
1	Unité Non Prête	Causes possibles : porte du lecteur de disquettes mal fermée, disquette incorrectement insérée, unité non connectée, disque dur n'ayant pas atteint sa vitesse opérationnelle.
2	Erreur	Le secteur adressé ne peut être lu ou écrit malgré plusieurs tentatives. Causes possibles : panne du coupleur, lecteur défectueux, dégradation du support.
3		Erreur de positionnement de la tête de lecture (supports sectorisés). Mêmes causes que pour l'erreur 2. Problèmes "hardware" : par exemple, tentative de lecture d'une disquette 600k sur lecteur 320k.
4	Accès Illégal	Référence à un secteur ou une piste hors des limites du support. Descripteur d'appel incorrect.
5	Protection Ecriture	Tentative d'écriture sur un support disque protégé physiquement en écriture. Carte entrée/sortie ou mémoire mal adressée ou défaut d'utilisation.
6 à 15		Erreurs spécifiques au périphérique.
16		Message d'erreur de SPR : écho caractère incorrect. L'écho reçu d'un caractère émis est différent de l'original.
17		Message d'erreur de SPR : incohérence logicielle. Code message incorrect. Zone de réception trop petite. Format de message incorrect. Cause possible : versions des logiciels utilisés incorrectes ou mal synchronisées entre elles ; écrasement du logiciel en mémoire.
18		Message d'erreur de SPR : réservé.
19		Message d'erreur de SPR : réservé.

Numéro du code-erreur	Libellé associé	Commentaires
20		Paramètre incorrect. Le numéro de tâche est égal à 0.
21		Réservé.
22		Tâche déjà créée. Tentative de création d'une tâche sous un numéro déjà attribué.
23		Tâche inconnue. Le numéro de tâche référencé dans la requête n'est pas attribué.
24		Table des descripteurs de tâches saturée. Aucun descripteur de tâche n'est disponible pour créer la tâche demandée.
25		Tâche déjà active. Tentative d'activation ou de suppression d'une tâche active.
26		Réservé.
27	Débordement Mémoire	L'espace mémoire disponible est insuffisant pour continuer le traitement en cours.
28		Tentative de libération d'un espace mémoire non alloué.
29		Tentative de libération d'un espace mémoire non alloué par votre poste.
30	Syntaxe Incorrecte	Erreur de syntaxe dans une commande. Désignation de fichier incorrecte. Séparateur, mot-clé incorrect.
31	Périphérique Inconnu	Le périphérique spécifié n'existe pas dans votre système.
32	Numéro d'unité	L'unité périphérique n'existe pas dans votre système.
33 à 35		Réservés.

Numéro du code-erreur	Libellé associé	Commentaires
36	Structure Fichier Incorrecte	La structure du fichier à exécuter est incorrecte. Le fichier à exécuter indiqué dans votre commande ne contient pas du code exécutable.
37		Réservé.
38	Débordement Mémoire	La taille du programme à exécuter est supérieure à la taille du plus grand segment mémoire disponible.
39		Réservé.
40	Fichier Inexistant	Le fichier spécifié est inexistant sur le support (spécifié ou pris par défaut).
41	Fichier Déjà Créé	Tentative de création d'un fichier dont le nom existe déjà dans le catalogue.
42	Fichier Non Ouvert	Tentative d'accès (par exemple : lecture ou écriture d'un fichier qui n'a pas été préalablement ouvert). Le numéro logique utilisé dans un descripteur ne correspond pas à un fichier ouvert.
43	Fichier Déjà Ouvert	Tentative de suppression d'un fichier non fermé (par un poste quelconque, éventuellement le sien).
44	Fichier Non Partageable	Tentative d'ouverture d'un fichier déjà ouvert en mode exclusif par un poste quelconque.
45	Trop de Blocs Extension	Pour un volume organisé en type 9F le nombre de blocs composant le fichier est déjà de 18. Il ne peut donc être étendu. Action : réorganiser le volume.
46	Débordement Volume	Tous les granules du volume sont occupés. Action : réduisez la longueur de certains fichiers ou supprimez-en. Utilisez un autre volume.

Numéro du code-erreur	Libellé associé	Commentaires
47	Clés d'Accès Invalides	Tentative d'accès à un fichier avec des clés incorrectes de lecture pour une lecture, de lecture et écriture pour une écriture ou une suppression.
48	Fin de Fichier	Détection d'une fin de fichier. Ce message n'est pas forcément une erreur mais une indication envoyée à l'utilisateur.
49	Catalogue Saturé	Il n'y a plus de place dans le répertoire (appelé aussi catalogue) du volume. Action : changez de volume, ou recréez celui-ci avec un nombre de fichiers maximum plus grand.
50		Le numéro logique fourni est égal à 0 ou supérieur au maximumm autorisé.
51	Trop de Fichiers NGF	Le nombre de fichiers ouverts simultanément par votre poste est trop grand.
52		Erreur numéro de poste. Le poste appelant n'est pas celui qui a ouvert le fichier.
53	Adresse Secteur Invalide	Tentative d'accès sur un secteur situé au-delà de la limite du fichier.
54	Structure Volume Incorrecte	Tentative d'utilisation d'un volume non créé par PROLOGUE.
55	Fonction Incorrecte	La fonction NGF spécifiée dans le descripteur est inconnue du système.
56	Fonction Non supportée	Fonction inexistante pour ce système.
57		Modification d'attribut impossible. Par exemple, demande de passage en mode exclusif, alors que le fichier est ouvert sous un autre poste ou un autre numéro logique.

Numéro du code-erreur	Libellé associé	Commentaires
58	Fonction Interdite	Tentative d'accès à une fonction supportée par le système mais interdite par la configuration.
59		Structure du volume incorrecte. Le nombre de granules du support ne peut être géré par le NGF.
60	Structure Fichier Incorrecte	L'instruction ou la fonction, qui s'adresse à des fichiers de type séquentiel, ne peut être utilisée pour ce fichier.
61		Réservé.
62		Perte d'information en écriture. La longueur des informations à écrire dans le fichier est supérieure à la longueur déclarée des enregistrements.
63		Perte d'information en lecture. La longueur de l'enregistrement à lire est supérieure à celle de la zone de réception.
64	Début de Fichier	Tentative de positionnement en deçà du début de fichier.
65 à 69		Réservés.
70		Longueur de clé trop petite dans le fichier séquentiel indexé.
71		Longueur de clé trop grande dans le fichier séquentiel indexé.
72	Trop de Fichiers SI	Trop de fichiers séquentiels indexés ouverts simultanément.
73		Index associé à la clé nul.
74		Clé déjà bloquée par un autre accès.
75	Débordement	Débordement du fichier Index. Fichier Index

Numéro du code-erreur	Libellé associé	Commentaires
76		Structure du fichier des clés incorrecte. Le fichier ne peut plus être exploité.
77	Débordement Fichier Données	Débordement du fichier des données.
78	Clé Inexistante	Clé non trouvée.
79		Clé trouvée mais index associé faux.
80		Trop de niveaux dans le graphe. Faites SVSI puis RTSI sur le fichier pour retrustructurer le fichier des clés.
81	Clé Déjà	Clé déjà existante. C'est une tentative d'insertion d'une clé déjà présente.
82		Taille du tampon de lecture plus petit que la clé.
83		Code-fonction inexistant ou numéro logique incorrect.
84		Trop de fichiers ouverts simultanément. Il faut alors modifier les paramètres de configuration du multicritère.
85		Incohérence du dictionnaire. Celui-ci est alors à recréer.
86		Trop de rubriques ou trop d'article dans le fichier multicritère.
87		Type de rubrique incorrect ou longueur de rubrique incorrecte ou nom de rubrique déjà existant ou lien déjà existant.
88		Description de rubrique inexistante où une rubrique figure plusieurs fois dans une question.

Numéro du code-erreur	Libellé associé	Commentaires
89		Nombre de critères incorrect ou trop de rubriques dans un lien ou trop de rubriques dans une interrogation.
90		Lien inexistant ou non créé.
91		Valeur multicritère trop longue.
92		Incompatibilité entre le fichier et le dictionnaire (par exemple, données modifiées sans passer par le multicritère).
93		Opérateur inconnu.
94		Type de description incorrect.
95		Réservé (multicritère).
96		Fin des réponses à une interrogation ou fin des rubriques ou fin des liens.
97		Expression multicritère incorrecte (BAL).
98		Utilisation d'une rubrique avant sa déclaration (cas d'un POSIT avant le KEY associé ...) ou rubrique non interrogeable (longueur > 255 octets).
99		Type de valeur incorrect.
100		Segment non existant. Tentative de faire référence à un segment non existant.
101		Fin de DATA pour une instruction READ. Il faut ajouter des éléments DATA ou utiliser l'instruction RESTORE.
102		Index zéro dans une table. Par exemple : B(0). Division par 0.
103		Index trop grand. Tentative d'utiliser une valeur d'index supérieure à la valeur maximale déclarée.

Numéro du code-erreur	Libellé associé	Commentaires
104		Chaîne utilisée comme numérique.
105		Retour incorrect après un GOSUB.
106		Débordement d'une variable numérique longue.
107		Trop de niveaux de GOSUB et/ou de LDGO.SEG. 24 niveaux maximum.
108		Débordement arithmétique.
109		Fichier intermédiaire incorrect. Erreur dans la structure d'un fichier "T-". Retraduire.
110		Variable numérique longue incorrecte. Liée à la structure du B.D.C.
111		Mauvais numéro de périphérique dans ASK ou PRINT. ASK = 1 est le seul numéro légal, PRINT = 1 ou PRINT = 2 conviennent. Tous les autres sont incorrects s'ils n'ont pas été déclarés auparavant par ASSIGN. Attention à la syntaxe PRINT = A ...
112		Erreur de format ou incohérence entre deux descriptions logiquement liées.
113		Variable trop grande. Ne peut être contenue dans la mémoire spécifiée.
114		Débordement de mémoire. Réviser le programme pour en faire des segments plus petits qui peuvent être appelés en mémoire en fonction des besoins.
115		Instruction inconnue. On peut avoir détérioré une partie de la mémoire.

Numéro du code-erreur	Libellé associé	Commentaires
116		Numéro logique non déclaré par un ASSIGN. Tentative d'utiliser une instruction se référant à un numéro logique, avant l'exécution d'un ASSIGN qui assigne un numéro au fichier spécifié.
117		Numéro logique se référant à un fichier, non à une unité de support. Dans une instruction IO, tentative de se référer au numéro logique d'un fichier.
118		Variable externe interdite à cet endroit de l'instruction.
119		Numéro logique se référant à une unité de support, non à un fichier. Dans une instruction qui utilise un fichier, on fait référence au numéro logique qui a été assigné à une unité périphérique.
120		Fichier ou support protégé contre l'écriture. Pour un fichier, fournir les clés d'accès. Pour un support, déprotéger celui-ci.
121		Le nom fourni est un nom d'unité et non un nom de fichier.
122		Mémoire tampon d'entrée trop petite. Dans la gestion des fichiers, la mémoire tampon d'entrée est trop petite pour contenir la clé ou l'index de l'enregistrement.
123		Pas de "ON ERROR" positionné lors de l'instruction "RESUME".
124		Pas d'erreur survenue lors de l'exécution d'une instruction "RESUME".
125		Instruction multicritère non gérée par cette version de l'exécuteur.

Numéro du code-erreur	Libellé associé	Commentaires
126		Fonction interdite sous cette méthode d'accès.
127		La variable que vous demandez de lire dans un fichier virtuel n'a jamais été écrite (c'est une erreur de fin de fichier en lecture).
170		Fonction inexistante (base de données).
171		Trop de fichiers ouverts simultanément.
172		Numéro logique incorrect ou descripteur incorrect ou mode d'ouverture incorrect.
173		Incohérence du dictionnaire.
174		Longueur de clé incorrecte ou longueur de données incorrecte ou type de fichier incorrect.
175		Description de jointure incorrecte. Nombre de joints binaires incorrect. Opérateur incorrect. Jointure sur le même fichier. Jointure impossible. Longueur/type de rubrique différents.
176		Description/nom de fichier existants.
177		Description/nom de fichier inexistants.
178		Nom de rubrique existant.
179		Nom de rubrique inexistant.
180		Description/nom de jointure existants.
181		Description/nom de jointure inexistants.
182		Trop de fichiers dans le dictionnaire. Trop de rubriques dans le dictionnaire. Trop de jointures dans le dictionnaire.
183		Pas de POSIT avant le DOWN ou UP.
184		Clé trouvée, mais index faux.

Numéro du code-erreur	Libellé associé	Commentaires
185		Pas de réponse pour ce fichier.
186 à 189		Réservés (Base de Données).
190		Descripteur environnement incorrect.
191		Table des numéros logiques saturée.
192		Appel incorrect.
193		Perte d'information en lecture.
194		Clôture non partageable.
195		Modification de clôture.
196		Fonction inexistante.
200		Fonction volume refusée (fichier ouvert sur ce volume).
201		Problème interne au NGF.
202		Incohérence entre le support et son image mémoire. Action : vérifiez l'intégrité de vos fichiers.
203 à 209		Réservés (NGF).
201 à 254		Réservés pour PROLOGUE.
255		Erreur dans un utilitaire PROLOGUE généralement appelé à partir d'un programme B.A.L.

C.1. STRUCTURE DES VOLUMES PROLOGUE GÉRÉS PAR LE NGF

Le NGF (noyau de gestion de fichier) est un module regroupant les fonctions élémentaires d'une gestion de fichiers sur supports sectorisés à accès aléatoire (disquette, disque dur fixe ou amovible).

Il offre à l'utilisateur, à partir d'une gestion dynamique de l'espace disponible sur support et d'un accès direct aux enregistrements, la possibilité de :

- définir le(s) type(s) d'organisation de ses fichiers,
- construire la (les) méthode(s) d'accès à ses enregistrements.

Dans tout ce qui suit, la notion de secteurs de 256 octets est utilisée.

Un volume PROLOGUE est nécessairement contenu sur une et une seule unité de ressource périphérique.

L'espace disque adressable par le NGF se divise en trois parties :

- La table d'allocation des granules unique et gérant l'ensemble de l'espace disque.
La description en est faite au paragraphe 2.5.3.
- Le catalogue du volume de longueur variable, défini à la création du volume, dont la description est faite dans les paragraphes ci-après.
- L'espace disque libre est utilisé pour l'allocation et désallocation dynamique de granules attribuée aux fichiers définis dans le catalogue.

C.1.1. Organisation d'un catalogue sous NGF

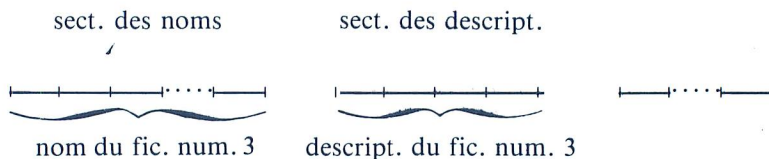
Le catalogue regroupe les noms des fichiers et leurs descripteurs.

Il commence au secteur absolu n° 4.

Il comprend des groupes de 9 secteurs décomposés comme suit :

- 1 secteur contenant les NOMs des fichiers :
 - chaque nom (ou début) occupe 8 octets
 - le secteur regroupe 32 NOMs de fichiers
- 8 secteurs contenant les descripteurs des fichiers dont les noms précèdent :
 - chaque descripteur occupe 64 octets
 - chaque secteur contient 4 descripteurs.

Les descripteurs sont ordonnés comme les noms :



Les huit premiers octets du secteur 4 (première place pour un nom) sont réservés pour mémoriser des informations du VOLUME. Il en est de même pour le descripteur associé (64 premiers octets du sect. 5).

La première entrée correspondant à un nom de fichier à zéro (8 octets à zéro binaire) indique la fin courante du catalogue. Lors d'une recherche de fichier, les entrées au-delà ne sont pas exploitées.

Informations pour le volume NGF

Elles occupent les huit premiers octets du catalogue (secteur 4, octets 0 à 7).

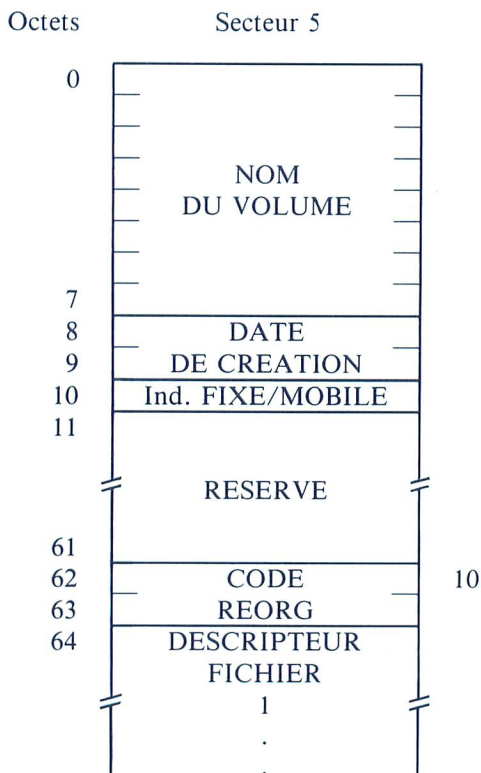
Elles permettent de définir :

- La taille des granules sur le volume (LGR) :
Cette taille est exprimée en secteurs de 256 octets.
Les valeurs possibles :
1 - 2 - 4 - 8 - 16 - 32 - 64 - 128 - 0 (c'est-à-dire 256)
- La taille du catalogue (LCAT) :
Cette taille, exprimée en secteurs représente le numéro du premier secteur physique n'appartenant pas au catalogue.
C'est le premier secteur disponible pour les données des fichiers.
Le nombre maximum de fichiers sur un volume NGF étant limité à 895, la taille maximum de LCAT est de 256 secteurs.
- L'identificateur du catalogue (IND) :
Cet indicatif permet au NGF de reconnaître un volume géré par lui
 - 9F hexa = Indicatif des supports correspondant à une version PROLOGUE 2.0 ou inférieure.
 - A0 hexa = Indicatif des supports correspondant à une version PROLOGUE 2.1 ou supérieure.
 - A1 hexa = La structure accepte les descripteurs d'extensions.
- L'indicateur volume copie (COP) :
 - = 0 : Volume NGF normal.
 - # 0 : Volume de type *COPIE*, obtenu par l'utilitaire "SV".
La valeur de l'indicateur donne le numéro d'ordre du volume dans la séquence, de 1 à 127.
Si le bit 7 est positionné, c'est le dernier volume de la copie.

FORMAT DU DESCRIPTEUR DE VOLUME

Le descripteur de volume complète les informations du secteur 4. Il est situé sur le secteur 5, octets 0 à 63.

Sa structure est la suivante :



La date de création a le format suivant :

0 a a a a a a m avec :

m m m j j j j j aaaaaa = Numéro d'année à partir de l'année 1984

mmmm = Numéro de mois de 1 à 12

jjjjj = Numéro de jour de 1 à 31.

Indicateur fixe/mobile.

00h = Volume amovible

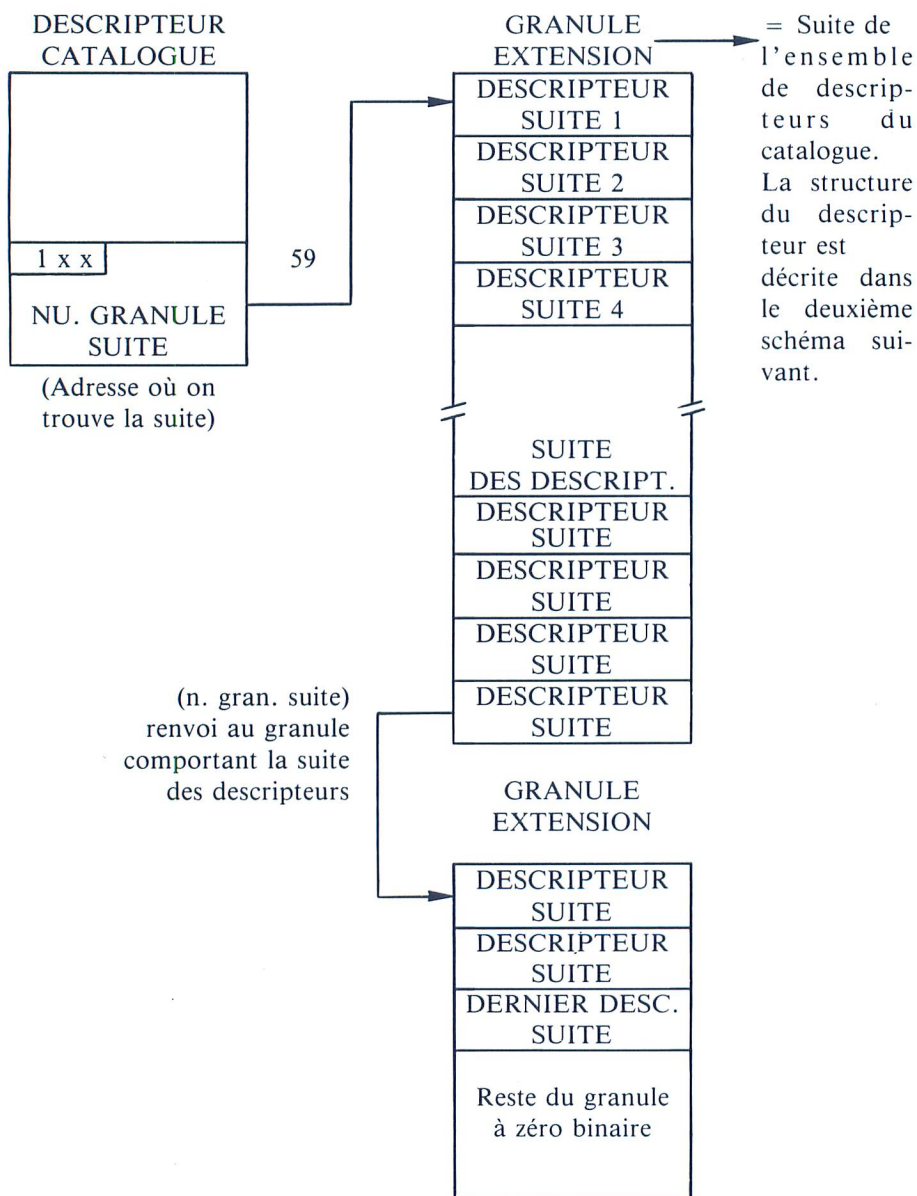
01h = Volume fixe.

Code REORG

Indique le numéro du granule utilisé par l'utilitaire REORG pour stocker ses tables de traitement.

Cette valeur doit normalement être à zéro. Elle n'est différente de 0 qu'en cours de réorganisation ou si la phase de réorganisation a été interrompue.

SCHEMA D'UNE CHAÎNE DE DESCRIPTEURS :



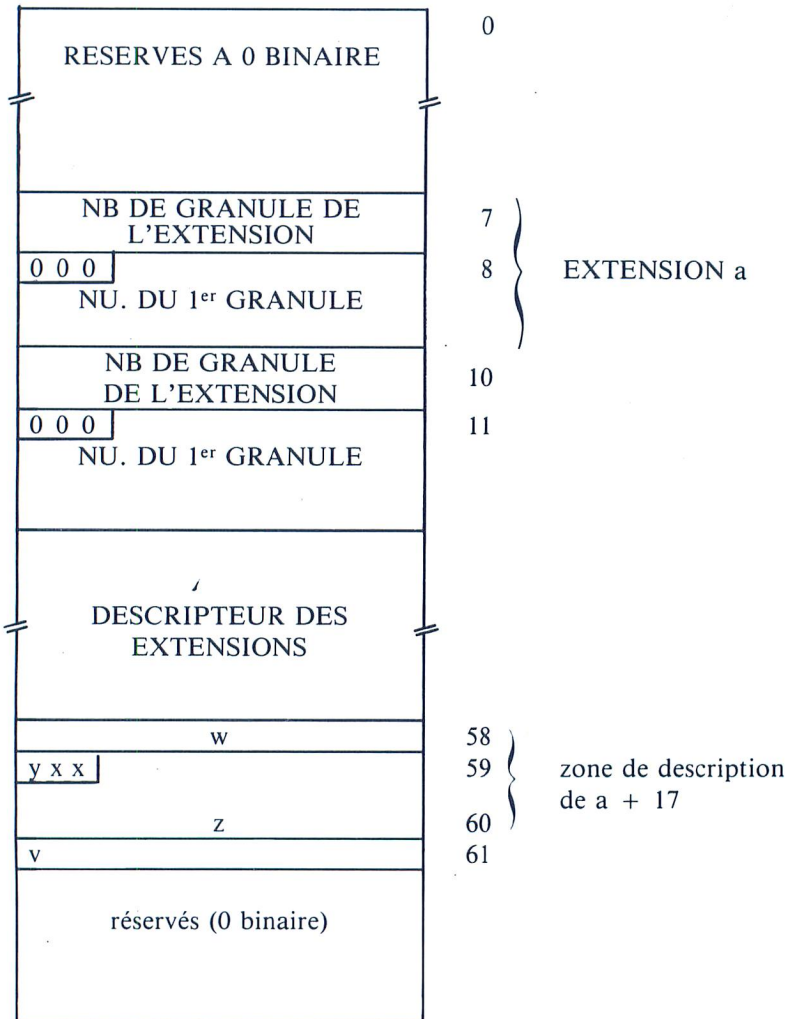
STRUCTURE SANS CHANGEMENT PAR RAPPORT AU PASSE	
w	
y x x	=
Structure sans changement par rapport au passé	

58 } zone pour la 18^e
59 } extension
60 }
61
62
63

bit	$y = 0$	w : nb de granules de la 18 ^e extension. z : nu. du premier granule de l'extension.
bit	$y = 1$	$w : 0$ z : nu. du granule extension.

C-7

STRUCTURE D'UN DESCRIPTEUR PRÉSENT DANS UN GRANULE EXTENSION



signification des zones en fonction de la position des bits "v" et "y".

bit	$v = 0$	ce descripteur possède un suivant.
bit	$v = 1$	ce descripteur est le dernier de la chaîne.
bit	$y = 0$	$w =$ nb. de granules de la 18 ^e extension de ce descripteur. $z =$ nu. du premier granule de l'extension.
bit	$y = 1$	$w = 0$. $z =$ nu. du granule suite.

STRUCTURE DISQUE DU CATALOGUE D'UN VOLUME NGF

Secteur 4 —>

0	lgr	lcat	ind	cop			
8	Nom du fichier 1						
16	Nom du fichier 2						

Secteur 5 —>

248	Nom du fichier 31
-----	-------------------

0	Descripteur VOLUME
---	-----------------------

64	Descripteur disque du fichier 1
----	------------------------------------

128	Descripteur disque du fichier 2
-----	--

192	Descripteur disque du fichier 3
-----	--

Secteur 6 —>

0	Descripteur disque du fichier 4
---	--

.....sect 7

.....sect 8

.....sect 10.....

		sect 11	
Secteur 13 —>		sect 12	
	0	Nom du fichier 32	
	8	Nom du fichier 33	
	16	Nom du fichier 34	
	:	:	:
Secteur 14 —>	248	Nom du fichier 63	
	0	Descripteur disque du fichier 32	
	64	Descripteur disque du fichier 33	
	128	Descripteur disque du fichier 34	
	192	Descripteur disque du fichier 35	
Secteur 15 —>	0	Descripteur disque du fichier 36	
	:	:	:
	:	:	:
		sect 16	
		sect 17	
		sect 18	
		sect 19	
		sect 20	
		sect 21	

ORGANISATION DISQUE DU CATALOGUE

SECTEURS	INFORMATIONS
0,1	Table d'allocation des granules
2,3	Extension de la table d'allocation des granules
4	Données du volume et noms des fichiers 1 à 31
5 à 12	Descripteur volume et descripteurs disques des fichiers 1 à 31
13 (*)	Noms des fichiers 32 à 63
14 à 21	Descripteurs disques des fichiers 32 à 63
22 (*)	Noms des fichiers 64 à 95
23 à 30	Descripteurs disques des fichiers 64 à 95
:	:
:	:
:	:
247	Noms des fichiers 864 à 895
248 à 255	Descripteurs disques des fichiers 864 à 895
256 (*)	

(*) Ces valeurs sont des longueurs possibles du catalogue

C.1.2. Mémorisation des noms des fichiers dans le catalogue NGF

La mémorisation des noms varie suivant leur structure.
Les structures possibles des noms sont les suivantes :

Structure 7/1 :

La partie principale du nom a une taille maximum de 7.
L'extension a une taille maximum de 1.

Ex. CP – X
partie principale : CP
extension : X

Structure 8/3 :

La partie principale du nom a une taille maximum de 8.
L'extension a une taille maximum de 3.

Ex. FICHER1 – EXE
partie principale : FICHER 1
extension : EXE

Mémorisation d'un nom structure 7/1 :

La partie principale est rangée dans les sept premiers octets de la zone NOM DE FICHER du catalogue.
L'extension est rangée dans le huitième octet de la zone.

Ex.

C	P							X
---	---	--	--	--	--	--	--	---

Mémorisation d'un nom structure 8/3 :

Les sept premiers octets de la partie principale sont rangés dans la zone NOM DE FICHIER du catalogue.

Le dernier octet et l'extension sont rangés dans le descripteur disque du fichier (voir descripteur).

EX.

01	F	I	C	H	I	E	R
----	---	---	---	---	---	---	---

L'indicatif 01 HEXADECIMAL permet de différencier les deux structures.

REMARQUES :

- Le NGF optimise la mémorisation : il essaie de mémoriser en 7/1 et s'il n'y parvient pas, il le mémorise en 8/3.
- La possibilité de créer des fichiers 8/3 est contrôlée par une autorisation d'exploitation donnée dans la configuration du noyau système.

DESCRIPTEUR DISQUE D'UN FICHIER STRUCTURE 7/1

—	Clés d'accès en LECTURE	—	0	
—	Clés d'accès en ECRITURE/DESTRUCTION	—	2	
—	RESERVES	—	4	
	Nombre de granules bloc 0		7	} Bloc 0
0 0 0	Numéro	—		
	du premier granule bloc 0			
	Nombre de granules bloc 1		10	} Bloc 1
0 0 0	Numéro	—		
	du premier granule bloc 1		13	
//				
	Nombre de granules bloc 17		58	} Bloc 17
0 0 0	Numéro	—	59	
	du premier granule bloc 17	—		
	Type du fichier		61	
—	RESERVES	—	62	

DESCRIPTEUR DISQUE D'UN FICHIER STRUCTURE 8/3

Fin du nom (p. principale)	0	
	1	
Extension du nom		
RESERVES	4	
Nombre de granules bloc 0	7	} Bloc 0
0 x x Numéro		
du premier granule bloc 0		
Nombre de granules bloc 1	10	} Bloc 1
y y y Numéro		
du premier granule bloc 1		
Nombre de granules bloc 2	13	
z z z Numéro		
du premier granule bloc 2		
	16	
Nombre de granules bloc 17	58	} Bloc 17
0 0 0 Numéro	59	
du premier granule bloc 17		
Type du fichier	61	
	62	
RESERVES		

Remarques :

- Les clés de protection sont réparties sur les 3 bits de poids fort des numéros des douze premiers blocs d'extension.
- 1 caractère de la clé a ses bits éclatés comme suit : 0 x x y y z z z
- La date de création/mise à jour d'un fichier est répartie sur les 3 bits de poids fort des numéros des 5 blocs suivants d'extension (blocs 12 à 16).

Les 16 bits de codification de la date se répartissent comme suit :

aaa	Bloc 12
aaa	Bloc 13
mmm	Bloc 14
mjj	Bloc 15
jjj	Bloc 16

- La date de création/mise à jour d'un fichier a la même structure que la date de création/mise à jour d'un volume.

Les versions de PROLOGUE, inférieures à 2.2, n'entretiennent pas la date de création/mise à jour des fichiers.

C.1.3. Gestion de l'espace disque sous NGF

Le NGF affranchit l'utilisateur des problèmes d'estimation préalable de l'espace disque nécessaire à ses fichiers.

Il alloue en effet à chaque fichier, au fur et à mesure de ses besoins, un quantum d'espace disque appelé 'GRANULE'.

Un GRANULE est un multiple de secteurs PROLOGUE de 256 octets, défini à l'avance par l'utilisateur, en fonction de la capacité du support utilisé et de la nature des fichiers à gérer (taille, taux d'expansion).

Ce paramètre (LGR), fixé par l'utilisateur au moment de la création du VOLUME, lui permet de rechercher une certaine optimisation dans la gestion de son espace disque.

Il est à noter que le paramètre LGR, une fois défini, est global pour l'ensemble des fichiers du volume et ne peut être modifié.

Table d'allocation d'un volume géré par NGF

Pour mémoriser l'allocation de l'espace disque à l'ensemble des fichiers d'un volume, le NGF dispose des quatre PREMIERS SECTEURS disque du volume (secteurs 0 à 3).

Les deux premiers octets de cette table représentent la capacité physique maximum de volume en granules (notation binaire inversée).

Un bit à 0 de la table des granules représentant un granule disponible, le nombre maximum de granules adressables par cette table est égal à **8 176 GRANULES**, ce qui autorise un espace adressable maximum théorique de 512 Moctets.

Suivant le nombre total de granules du volume, la table d'allocation peut occuper 1 à 4 secteurs :

Nombre de granules	Nombre de secteurs
< = 2032	1
< = 4080	2
< = 6128	3
< = 8176	4

C'est lors de la création du volume, et en fonction des deux paramètres :

- taille des granules,
- capacité du support,

que le système détermine le nombre de secteurs à allouer pour la table des granules.

Les systèmes de version inférieure à 2.2. ne peuvent reconnaître une table des granules de taille supérieure à 2 secteurs.

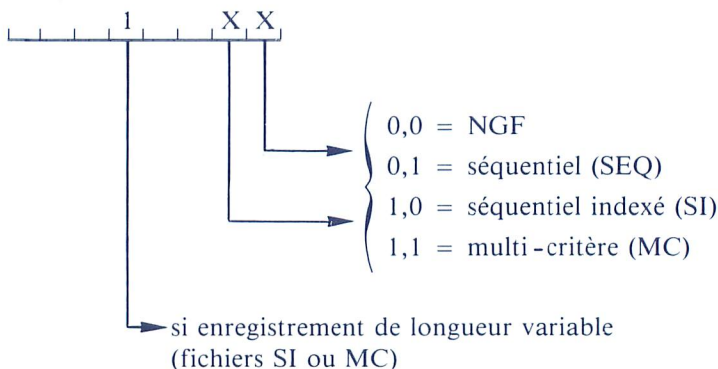
La table d'allocation du volume, gérée en simultanéité avec le descripteur disque d'un fichier, permet au NGF de suivre l'évolution de ce fichier sur le volume.

C.1.4. Remarques complémentaires

Octet type de fichier

Octet 61 des descripteurs DISQUE et MÉMOIRE

Cet octet permet de reconnaître le type d'organisation du fichier en question : séquentiel, séquentiel indexé, etc.



Utilisation des zones réservées

Les octets 4, 5, 6 et 62, 63 du descripteur disque définissent deux zones réservées à des natures de fichiers particulières.

Dans le cas d'un fichier NGF simple, ces zones sont à zéro binaire.

— Cas d'un fichier SÉQUENTIEL :

Octets 4, 5 Longueur des enregistrements

Octet 6 Réservé à zéro

Octets 62, 63 Pointeur de fin de fichier

— Cas d'un fichier SÉQUENTIEL INDEXÉ

Octets 4, 5 Longueur de l'article

Octet 6 Longueur de la clé

Octets 62, 63 Réservés à zéro

— Cas d'un fichier sous émulateur CP/M ou MS-DOS

Octet 4 Adresse du premier octet libre dans le dernier secteur occupé du fichier

Octets 5, 6 Adresse du dernier secteur de 256 occupé

Octets 62, 63 Réservés à zéro binaire.

ANNEXE D : LISTE DES REFERENCES

SRS 1919 PROLOGUE avec Multivue pour Micral 30

SRS 1919 PROLOGUE avec Multivue pour Micral 60

SRS 1929 PROLOGUE avec Multivue pour Micral 90.20

SRS 1959 PROLOGUE avec Multivue pour Micral 90.50

ANNEXE E : BIBLIOGRAPHIE

BIBLIOGRAPHIE DISPONIBLE CHEZ BULL

Ouvrages

Références

– Guide Utilisateur du Micral xxxx

Ce manuel vous offre toutes les informations utiles à la bonne utilisation du micro-ordinateur. Il en existe un par modèle de Micral.

– Manuel de référence Technique du Micral xxxx

Ce guide, pour informaticiens, vous donne une connaissance exhaustive du Système. Il existe un manuel par modèle de Micral.

– Manuel de maintenance du Micral xxxx

Ce manuel réservé aux techniciens de maintenance permet de dépanner le micro-ordinateur. Il existe un manuel par modèle de Micral.

– Schémathèque du Micral xxxx

Une Schémathèque par modèle de Micral.

– Catalogue des pièces de rechanges du Micral xxxx

Un catalogue pour chaque Micral.

– Guide de mise en œuvre du coprocesseur 80287

Ce manuel vous explique le fonctionnement et l'installation d'un coprocesseur mathématique 80287. C'est une option installable sur certains modèles.

05 F1 15 MK
05 A1 15 MK

- Guide de mise en œuvre du disque 10 Mo** 05 F1 16 MK
05 A1 16 MK

Ce manuel décrit l'installation et le fonctionnement du disque de capacité 20 Mo. C'est une option de certains modèles.
- Guide de mise en œuvre du disque 40 Mo** 05 F1 17 MK
05 A1 17 MK

Ce manuel décrit l'installation et le fonctionnement du disque de capacité 40 Mo. C'est une option de certains modèles.
- Guide de mise en œuvre de la disquette 1,2 Mo** 05 F1 18 MK
05 A1 18 MK

Ce manuel décrit l'installation et le fonctionnement de la disquette de 1,2 Mo. C'est une option de certains modèles.
- Guide de mise en œuvre de la disquette 360 Ko** 05 F1 19 MK
05 A1 19 MK

Ce manuel décrit l'installation et le fonctionnement de la disquette de 360 Ko. C'est une option de certains modèles.
- Guide de mise en œuvre du dérouleur de bande 20 Mo intégré** 05 F1 20 MK
05 A1 20 Mk

Ce manuel décrit l'installation et le fonctionnement du dérouleur de bande de capacité 20 Mo. C'est une option de certains modèles.
- Guide de mise en œuvre du coffret du dérouleur de bande 20 Mo** 05 F1 21 MK
05 A1 21 MK

Même manuel que le précédent, mais le dérouleur est en coffret au lieu d'être intégré.
- Guide de mise en œuvre de la carte mémoire 1 Mo** 05 F1 23 MK
05 A1 23 MK

Ce manuel décrit l'installation et le fonctionnement de la carte extension 1 Mo. C'est une option de certains modèles.

- Guide de mise en œuvre de la carte mémoire 128 Ko**

Ce manuel décrit l'installation et le fonctionnement de la carte mémoire 128 Ko. C'est une option de certains modèles.

05 F1 24 MK
05 A1 24 MK
- Guide de mise en œuvre de la carte vidéo couleur**

Ce manuel décrit l'installation et le fonctionnement de la carte vidéo couleur. C'est une option de certains modèles.

05 F1 25 MK
05 A1 25 MK
- Guide de mise en œuvre de l'écran couleur**

Ce manuel décrit l'installation et le fonctionnement de l'écran couleur, option de certains modèles.

05 F1 26 MK
05 A1 26 MK
- Guide de mise en œuvre de la carte série parallèle**

Ce manuel décrit l'installation et le fonctionnement de la carte série parallèle, option de certains modèles.

05 F1 28 MK
05 A1 28 MK
- Guide de mise en œuvre de la carte multivoie**

Ce manuel décrit l'installation et le fonctionnement de la carte 4 voies V24, option de certains modèles.

05 F1 29 MK
05 A1 29 MK
- Guide de mise en œuvre de la carte de communication une voie asynchrone**

Ce manuel décrit l'installation et le fonctionnement de la carte de communication 1 voie V24, option de certains modèles.

05 F1 30 MK
05 A1 30 MK
- Guide de mise en œuvre de la carte de communication synchrone**

Ce manuel décrit l'installation et le fonctionnement de la carte de communication synchrone, option de certains modèles.

05 F1 31 MK
05 A1 31 MK

– Guide de mise en œuvre de la carte de communication HDLC

05 F1 32 MK
05 A1 32 MK

Ce manuel décrit l'installation et le fonctionnement de la carte de communication HDLC, option de certains modèles.

– Guide de mise en œuvre de piedestal pour unité centrale verticale

05 F1 33 MK
05 A1 33 MK

Ce manuel décrit l'installation du pied amovible, option de certains modèles.

– Guide de mise en œuvre de l'écran-clavier pour Multipostes

05 F1 34 MK
05 A1 34 MK

Manuel d'installation et de fonctionnement d'un poste écran/clavier 100 touches.

– Guide de mise en œuvre de l'écran monochrome

05 F1 27 MK
05 A1 27 MK

Manuel d'installation et de fonctionnement de l'écran monochrome, option de certains modèles.

– Guide de mise en œuvre de la carte monochrome graphique

05 F1 89 MI
05 A1 89 MI

Manuel d'installation et de fonctionnement de la carte monochrome graphique, option de certains modèles.

– Guide utilisateur des imprimantes PRT 1910-1911-1912

05 XI 35 MI

Manuel d'utilisation des imprimantes PRT 1910, 1911, et 1912, comprenant les codes de programmation et l'utilisation sur tous les modèles de la gamme.

– Guide utilisateur des imprimantes PRT 1913-1914

05 X1 46 MK

Manuel d'utilisation des imprimantes PRT 1913, 1914, comprenant les codes de programmation et l'utilisation sur tous les modèles de la gamme.

- Guide utilisateur des imprimantes PRT 1241** 05 X1 47 MK

Manuel d'utilisation des imprimantes PRT 1241
comprenant les programmation et l'utilisation sur
tous les modèles de la gamme.
- Guide d'installation de PROLOGUE sur Micral compatible**

Manuel décrivant les fonctions spécifiques de 05 F2 09 NA
PROLOGUE pour votre modèle de 05 A2 09 NA
micro-ordinateurs. 05 D2 09 NA
- Guide de l'Utilisateur de PROLOGUE** 05 F2 9663

Manuel décrivant les fonctionnalités générales de 05 A2 9663
PROLOGUE. 05 D2 9663
- Guide du programmeur PROLOGUE** 05 F2 11 MI

Manuel décrivant le système PROLOGUE et ses 05 A2 11 MI
fonctions d'interface système avec les micro
processeur 16 bits.
- Guide de l'Utilisateur MS-DOS** 05 F2 26 MI

Manuel décrivant les fonctions du système 05 A2 26 MI
d'exploitation MS-DOS. 05 D2 26 MI
05 I2 26 MI
- Guide du programmeur MS-DOS** 05 F2 42 MI

Manuel décrivant le système MS-DOS et ses 05 A2 42 MI
fonctions d'interface avec les microprocesseurs
16 bits.
- Memento MS-DOS** 05 F2 61 MI

05 A2 61 MI
- Manuel de référence GW-BASIC** 05 F2 34 MI

Manuel décrivant les instructions du langage 05 A2 34 MI
GW-BASIC

— Memento GW-BASIC		05 F2 02 MK 05 A2 02 MK
— Manuel de référence B.A.L.		05 F2 9664 05 A2 9664
Manuel décrivant les instructions du langage B.A.L.		
— Manuel de référence B.A.L. Graphique		05 F2 37 MI 05 A2 37 MI
Manuel décrivant les instructions graphiques du langage B.A.L.		
— Manuel de référence B.A.L. - SGBD		05 F2 23 MI 05 A2 23 MI
Manuel décrivant la base de données relationnelle fonctionnant avec le langage B.A.L.		
— Manuel de référence COBOL	Tome 1	05 A2 18 MI
	Tome 2	05 A2 63 MI
Manuels décrivant les instructions du langage COBOL.		
— Manuel de référence FORTRAN		05 A2 16 MI
Manuel décrivant les instructions du langage FORTRAN.		
— Manuel de référence PASCAL	Tome 1	05 A2 50 MI
	Tome 2	05 A2 51 MI
Manuels décrivant les instructions du langage PASCAL ISO		
— Memento PASCAL		05 A2 72 MI
— Langage d'Assemblage 8086/8088		05 F2 55 MI 05 A2 55 MI
Manuel décrivant le langage d'assemblage des micro processeurs 16 bits de la série INTEL 86.		
— Macro-assembleur MASM		05 F2 20 MI 05 A2 20 MI
Manuel décrivant les macro-instructions d'assemblage pour les processeurs de la série INTEL 86.		

- **Manuel de référence FORTRAN**

Manuel décrivant les instructions du langage FORTRAN.

05 F2 16 MI

05 A2 16 MI

- **Memento FORTRAN**

05 F2 41 MK

05 A2 41 MK

- **Manuel de référence PASCAL**

Manuels décrivant les instructions du langage PASCAL ISO

Tome 1 05 F2 50 MI

05 A2 50 MI

05 F2 51 MI

05 A2 51 MI

- **Memento PASCAL**

05 F2 72 MI

05 A2 72 MI

- **Langage d'Assemblage ASM286**

Manuel décrivant le langage d'assemblage des micro processeurs 16 bits de la série INTEL 286.

05 F2 43 MK

05 A2 43 MK

- **Macro-assembleur MAS M**

Manuel décrivant les macro-instructions d'assemblage pour les processeurs de la série INTEL 286.

05 F2 44 MK

05 A2 44 MK

- **Manuel de mise en œuvre SPR**

Manuel décrivant l'installation, la configuration et l'utilisation du réseau SPR.

05 F2 36 MK

05 A2 36 MK

- **Guide de mise en œuvre du réseau TEN-NET**

Manuel décrivant l'installation, la configuration et l'utilisation du réseau TEN-NET.

05 F2 38 MK

05 A2 38 MK

Ces ouvrages sont fournis avec le produit auquel ils se rapportent ou peuvent être achetés au centre de documentation BULL :

CEDOC

Parc Industriel d'Incarville

BP 110

27100 VAL de REUIL



LEXIQUE

LEXIQUE :

Dictionnaire succinct d'une science ou d'une technique, d'un domaine spécialisé.

Avertissement :

Ce document ne prétend pas définir tous les termes informatiques en usage aujourd'hui. Il se propose uniquement de reprendre les principaux mots-clés rencontrés dans l'ensemble de la Documentation BULL MICRAL.

A

Accès direct (Logiciels de base)

Random access

Mode d'exploitation d'un fichier dont les articles peuvent être présentés au moyen d'une adresse, dans un ordre indépendant de leur position sur le support.

On dit aussi : **accès relatif, accès aléatoire.**

Accès séquentiel (Logiciels de base)

Serial access

Mode d'exploitation d'un fichier dont les articles, disposés physiquement sur le support dans l'ordre de présentation du fichier, ne peuvent être lus qu'en suivant cet ordre.

Accès séquentiel indexé (Logiciels de base)

Pour accéder à une information, on utilise un index qui indique l'adresse de l'enregistrement recherché. Cette méthode combine les avantages de l'accès direct (plus rapide) et de l'accès séquentiel (encombrement réduit).

Voir également : **Multi-critères.**

Adaptateur (Télécom)

Adapter

Dispositif de liaison entre deux appareils de caractéristiques différentes.

Adresse (Termes fondamentaux)

Address

Symbole permettant de repérer l'emplacement d'un registre d'une zone mémoire.

Adresser (Termes fondamentaux)

To address

Faire référence à une donnée par son adresse.

Algorithme (Analyse)

Algorithm

Ensemble de règles destiné à résoudre un problème au moyen d'un nombre fini d'opérations.

Alphanumérique (Programmation)

Alphanumeric

Relatif à un jeu de caractères qui comprend à la fois des lettres, des chiffres, et éventuellement des caractères spéciaux.

Alternat (Télécom)*Half duplex*

On parle de transmission de données à l'alternat lorsque les données sont transmises dans un seul sens à la fois.

On dit aussi : **semi-duplex**.

Amorce (Logiciels de base)*Bootstrap*

Programme qui se charge lui-même.

Analogique (Termes fondamentaux)*Analog*

Par opposition à **numérique** ; un signal analogique d'une information relative à une grandeur respecte la nature continue de grandeur en reproduisant, par analogie, cette information sous une autre forme.

Anneau (Télécom)*Ring*

Topologie de réseau dans laquelle les sites sont raccordés les uns aux autres sous forme d'une boucle.

Appeler (Programmation)*To call*

Lancer un programme ou un sous-programme situé à une adresse donnée

Application (Termes fondamentaux)*Application*

Ensemble des procédures, fichiers, programmes mis au point pour une utilisation déterminée.

Argument (Termes fondamentaux)*Argument*

Désigne une donnée, servant d'indicatif, de clé ou d'étiquette lors d'un tri, ou lors d'une recherche dans une table ou un fichier.

Arrière-plan (Graphique)*Background*

1. Zone qui entoure le sujet.
2. Fond sur lequel est écrit un caractère.

Article (Gestion de fichiers)*Item*

Ensemble organisé de données se rapportant à un même objet ou à un même sujet.

ASCII

American Standard Code for Information Interchange.

Code normalisé utilisé pour l'échange d'informations sur les systèmes de traitement de données et l'équipement associé. Un fichier ASCII est un fichier texte dans lequel les caractères sont représentés par des codes ASCII.

ASSEMBLEUR (Langages)

Assembler, Assembly language

Langage, proche d'un langage de type machine. L'assembleur traduit le code assembleur en code binaire exécutable par la machine.

Asynchrone (Télécom)

Asynchronous

Ce mode de communication permet de véhiculer un caractère à la fois, chaque caractère étant encadré par des bits de signalisation. Souvent utilisée entre terminaux et unités centrales, sa vitesse est moindre que celle d'une communication **synchrone**.

Voir également : **synchrone**.

Attribut

Attribute

1. (Programmation)

Complément d'information explicité servant à décrire les caractéristiques d'une donnée.

On dit aussi : **paramètre**.

2. (Visualisation)

Fonction spécifique exécutée par l'affectation de codes hexadécimaux à l'écran.

Avant-plan (Graphique)

Foreground

Partie de l'affichage correspondant au caractère proprement dit.

B

B.A.L. (Langages)

Business Basic Application Language.

Langage de programmation très complet développé sous le système d'exploitation PROLOGUE. Il permet des applications aussi variées que la gestion de fichiers, le graphique, les télécommunications, l'utilisation de bases de données.

Banque de données (Gestion de fichiers)

Data bank

Rassemblement cohérent de tous les fichiers apparentés, d'origines diverses, mis à la disposition de plusieurs utilisateurs pouvant les exploiter en fonction de leurs besoins propres.

Bascule (Matériels)

Flip-flop, Trigger

Dispositif présentant deux états stables, capable de passer alternativement de l'un à l'autre sous l'effet d'une impulsion appropriée.

On dit aussi : **bistable**.

Base de données (Gestion de fichiers)

Data base

Ensemble d'informations organisées en un fichier unique de telle façon que leur gestion (par un système de gestion de base de données ou S.G.B.D.) soit indépendante des applications auxquelles elles servent.

Voir également : **S.G.B.D.**

BASIC (Langages)

Beginners All-purpose Symbolic Instruction Code.

Langage de programmation évolué. C'est le plus répandu sur les micro-ordinateurs.

Voir également : **GW-BASIC**.

Baud (Télécom)

Baud

Unité de vitesse de modulation.

Bibliothèque (Logiciels de base)

Library

Ensemble organisé des programmes enregistrés sur un support externe, correspondant aux besoins de l'exploitation.

Binaire (Termes fondamentaux)

Binary

Concerne la numération en base 2, c'est-à-dire n'utilisant que les chiffres 0 et 1.

Bit (Termes fondamentaux)

Binary digit

Chiffre binaire, 0 ou 1. C'est le plus petit élément d'information.

Bogue (Programmation)

Bug

Voir : **Erreur**

Boucle (Programmation)

Loop

Ensemble d'instructions qui peut être exécuté plusieurs fois tant qu'une condition reste vraie.

Branchement (Programmation)

Branch

Opération par laquelle l'ordre d'exécution des instructions peut être modifié.

Branchement conditionnel (Programmation)

Conditional branch

Branchement qui entraîne l'exécution d'une séquence d'instructions parmi plusieurs séquences, en fonction de l'analyse d'un critère logique.

Branchement inconditionnel (Programmation)

Unconditional branch

Branchement qui impose l'exécution d'une séquence d'instructions qui n'est pas écrite immédiatement après l'instruction de branchement.

Bus (Matériels)

Bus

Liaison électrique entre les différents éléments d'un micro-ordinateur.

On distingue :

- les bus de commandes qui transmettent des messages d'opérations.
- les bus d'adresses qui transmettent les adresses cases-mémoire.
- les bus de données qui transmettent le contenu des cases-mémoire et le résultat des calculs.

C

Canal (Télécom)

Channel

Voie de transmission des signaux.

Capacité de mémoire (Termes fondamentaux)

Memory capacity

Nombre d'éléments binaires, de caractères, d'octets ou de mots qu'une mémoire peut contenir.

Caractère (Termes fondamentaux)

Character

Lettre, chiffre ou autre symbole utilisé dans l'organisation ou la représentation de données. Une séquence continue de caractères s'appelle chaîne de caractères.

Carte (Matériels)

Card

Circuit imprimé supportant un ou plusieurs circuits intégrés.

Exemples : cartes graphiques, d'extension mémoire, de communication.

Chaîne (Termes fondamentaux)

String

Suite linéaire d'éléments tels que des caractères.

Chemin (Logiciels de base)

Path

Dans le cas de la recherche d'un sous-répertoire (ou d'un fichier), désigne les différentes étapes à franchir pour accéder à ce sous-répertoire, en progressant de la racine vers ce sous-répertoire.

Voir également : **répertoire hiérarchisé**.

Circuit intégré (Matériels)

Chip

Circuit électronique miniaturisé implanté sur une plaque de silicium.

Clé (Termes fondamentaux)

Key

Suite de chiffres binaires ou de caractères faisant partie d'un ensemble de données, qui caractérisent cet ensemble et permettent de l'identifier. Se différencie de l'**étiquette**, qui est extérieure à l'ensemble de données auquel elle se rapporte.

Clôture (Graphique)*View, Viewport*

Concerne les applications graphiques dans certains langages de programmation, tels GW-BASIC et B.A.L.

Dans l'espace écran indépendant de l'espace de coordonnées utilisateur, la clôture délimite une partie rectangulaire, de surface inférieure ou égale à la surface de l'espace écran.

Voir : **fenêtre**.

COBOL (Langages)

COMmon **B**usiness **O**riented **L**anguage.

Langage évolué orienté vers les applications de gestion.

Commande (Termes fondamentaux)*Command*

Opération élémentaire permettant de déclencher une action du système d'exploitation.

Voir également : **Interpréteur de commandes**.

Communications (Télécom)*Data transmission*

Emission et réception de données.

Commutation de circuits (Télécom)*Circuit switching*

En télécommunications, technique dans laquelle, pendant toute la durée d'une transmission, il est établi entre deux correspondants un circuit permanent.

Compilateur (Langages)*Compiler*

Programme qui transforme un programme écrit en langage évolué en un programme exprimé en langage proche de la machine.

Concaténation (Termes fondamentaux)*Concatenation*

Opération qui lie deux chaînes de caractères dans l'ordre spécifié, formant ainsi une seule chaîne de caractères d'une longueur égale à la somme des longueurs des deux chaînes.

Configuration

Configuration

1. (Matériels).
Composition physique d'un micro-ordinateur, précisée par la nature, le nombre et les caractéristiques essentielles de ses éléments.
2. (Logiciels de base)
Incorporation des modules de contrôle des périphériques.
3. (Logiciels de base)
Action (ou résultat de l'action) consistant à définir les modules nécessaires à l'exploitation.

Constante (Termes fondamentaux)

Constant

Donnée dont la valeur est prédéterminée et ne varie pas durant l'exécution d'un programme.

Contrôleur de périphériques (Matériels)

Handler

Elément d'un ordinateur qui gère le fonctionnement d'un périphérique dont il est spécifique, sous la commande du canal auquel il est connecté ou de l'organe de commande ; il adapte en outre les caractéristiques des signaux reçus à celles de l'organe auquel il les transmet.

Conversationalnel (Termes fondamentaux)

Interactive

Voir **Interactif**.

Convivial (Termes fondamentaux)

Qualifie un outil informatique dont l'utilisation est simplifiée au maximum.

Un système convivial est conçu pour permettre une utilisation aisée par un non-informaticien.

Coordonnée absolue (Graphique)

Absolute coordinate

En mode graphique, spécification de l'adresse d'un point par rapport à un système de coordonnées.

Coordonnée relative (Graphique)

Relative coordinate

En mode graphique, valeur qui identifie l'emplacement d'un point en spécifiant les déplacements à partir d'un autre point.

CP/M (Logiciels de base)

Control Program for Microprocessors.

Système d'exploitation, créé par Digital Research, très utilisé sur les micro-ordinateurs à processeurs 8 bits.

Crayon optique (Matériels)

Lightpen

Stylo électronique, sensible à la lumière, relié par fil au micro-ordinateur, permettant de désigner sur l'écran des points ou des options d'un menu sans avoir recours au clavier. Il permet donc de réaliser des dessins directement sur écran.

On dit aussi : **photostyle**.

Cylindre (Matériels)

Cylinder

Ensemble des pistes d'une mémoire à disques magnétiques qui peuvent être desservies par un groupe solidaire de têtes de lecture, sans déplacement de celui-ci, lorsque l'une des têtes a été placée face à une piste.

Une unité de disques à têtes fixes comporte un seul cylindre.

Voir également : **piste**.

D

Débordement de mémoire (Termes fondamentaux) *Memory overflow*

Phénomène résultant d'une tentative d'enregistrement d'une donnée (le plus souvent le résultat d'une opération) dans une zone de mémoire ou un registre de longueur inférieure à celle de cette donnée.

On dit aussi : **dépassement de capacité**.

Débranchement (Programmation) *Trapping*

Rupture dans le déroulement normal d'une opération, permettant de reprendre plus tard l'opération au point où elle s'était arrêtée et généralement provoquée par un signal d'origine externe.

On dit aussi : **déroutement**.

Décalage (Termes fondamentaux) *Shift*

Déplacement simultané de caractères ou d'éléments binaires d'une donnée, d'une ou plusieurs positions vers la droite ou vers la gauche.

Décor (Logiciels de base)
Programme utilitaire permettant d'exécuter sous PROLOGUE des logiciels et des progiciels conçus pour être mis en œuvre avec d'autres systèmes d'exploitation de micro-ordinateurs.

Dépassement de capacité (Termes fondamentaux)

Voir : **Débordement de mémoire**.

Dérouleuse de bande (Matériels) *Magnetic tape drive, Streamer*

1. Appareil servant à entraîner la bande magnétique, en permettant la lecture et l'écriture.
2. Dispositif conçu pour permettre la sauvegarde de données exploitées sur disque.

Déroutement (Programmation) *Trapping*

Voir : **Débranchement**.

Diagnostic (Termes fondamentaux) *Diagnostic*

Détection et isolation d'un mauvais fonctionnement ou d'une erreur.

DIALOGUE (Logiciels de base)

Progiciel interactif. **Dialogue** est supporté par le système d'exploitation **Prologue**. Il est orienté vers la manipulation de données et repose sur une base de données qui gère des fichiers de type multi-critères.

Didacticiel (Logiciels de base)*Courseware*

Logiciel éducatif.

Disque dur (Matériels)*Disk*

Support magnétique permettant de stocker des informations. A progressivement remplacé la bande magnétique. De capacité nettement supérieure aux disquettes, les disques durs sont intégrés dans les systèmes à vocation professionnelle, en attendant les disques optiques.

Disquette (Matériels)*Floppy disk*

Support magnétique permettant de stocker des informations. Une disquette est un disque amovible souple, très facile à manipuler. Elle constitue le type de support le plus répandu en micro-informatique. On distingue les disquettes 8 pouces, 5 pouces 1/4 et les micro-disquettes 3 pouces 1/2.

On dit aussi : **minidisque**.

Duplex intégral (Télécom)*Full duplex*

Mode de transmission s'effectuant dans les deux sens à la fois.

On dit aussi : **transmission bidirectionnelle simultanée**.

E

E.B.C.D.I.C. (Abréviation)

Extended Binary Coded Decimal Interchange Code.

Code de caractères à 8 éléments binaires, adopté par plusieurs constructeurs comme code interne des ordinateurs et utilisé également comme code externe.

Echo (Termes fondamentaux)

1. Confirmation à l'émetteur de la réception de ses données.
2. Renvoi vers l'imprimante ou l'écran d'un caractère introduit au clavier.

Editeur (Logiciels de base)

Editor

Programme spécialisé offrant différents modes de présentation.
Editeur de textes, graphique, d'images...

Editeur de liens (Logiciels de base)

Link editor

Logiciel chargé de relier plusieurs tronçons de programme en langage machine, en y adjoignant éventuellement des routines extraites de la bibliothèque, pour constituer un programme exécutable.

Emulateur (Termes fondamentaux)

Emulator

Dispositif généralement associé à un logiciel spécial, spécifique d'un type d'ordinateur, destiné à transformer dans son langage propre et à adapter à son organisation interne, un programme rédigé dans le langage d'un autre type d'ordinateur.

Entrée (Termes fondamentaux)

Input

Réception de données externes par le micro-ordinateur, ou l'un de ses éléments.

On dit aussi : **introduction**.

Entrée/Sortie (Termes fondamentaux)

Input/Output

Echange d'informations avec l'extérieur.

On parle de dispositifs d'entrée (clavier) et de dispositifs de sortie (imprimante, synthétiseur vocal).

Erreur (Programmation)

Bug

Anomalie se situant au niveau de la conception ou de l'exécution qui interrompt le déroulement d'un travail.

Etiquette (Programmation)*Label*

1. Suite de caractères liée à un ensemble de données, qui le caractérise et permet de l'identifier.
2. Nom symbolique d'une adresse dans un programme en langage évolué.

Etoile (Télécom)*Star*

Topologie de réseau dans laquelle chaque site est relié au site central.

F**Fenêtre (Graphique)***Window*

Concerne les applications graphiques dans certains langages de programmation, tels GW-BASIC et B.A.L.

Dans l'espace de coordonnées utilisateur, la fenêtre délimite une partie rectangulaire de côtés parallèles aux axes du système de coordonnées utilisateur.

Voir : **clôture**.

Fichier (Logiciels de base)*File*

Ensemble organisé d'articles ou d'enregistrements de même nature susceptibles de faire l'objet de traitements définis, ou issus de tels traitements.

Format (Termes fondamentaux)*Format*

Forme et organisation d'une information sur un support. Il est indépendant de la représentation codée de l'information.

Formatage (Termes fondamentaux)*Formatting*

Opération qui consiste à inscrire (à l'aide du système d'exploitation) des repères délimitant des zones d'enregistrement sur une disquette. On parle aussi de **prémarquage** (PROLOGUE).

FORTRAN (Langages)

FORMula **TRAN**slating system.

Langage évolué orienté vers les applications scientifiques.

Fusion (Programmation)*Merge*

Réunion de plusieurs fichiers en un seul.

G

Go (Abréviation)

Giga-octet. Vaut 2^{30} octets.

Voir également : **octet**.

Granule (Logiciels de base)

Dans le cadre de l'organisation de Fichiers PROLOGUE, unité logique de stockage sur un support sectorisé. Il correspond à un certain nombre de secteurs sur un volume, nombre défini à la création du volume. (obligatoirement une puissance de 2. Par défaut, 16).

GW-BASIC (Langages)

BASIC graphique.

H

Horloge (Matériels)

Clock, Timer

Source d'impulsions régulièrement espacées dans le temps et utilisée :

- soit pour rendre synchrone le fonctionnement d'un ou de plusieurs organes d'un ordinateur.
- soit pour mesurer des intervalles de temps par comptage d'impulsions dans un registre spécial, dit registre d'horloge.

Hôte (Télécom)

Host

Ordinateur vers lequel convergent plusieurs lignes d'un réseau et où se trouvent concentrées des activités intéressant les sites connectés, ordinateurs satellites ou terminaux : gestion d'une base de données, calculs demandant une puissance importante de traitement...

I

Imprimante (Matériels)

Printer

Périphérique matérialisant sur papier les résultats fournis par un ordinateur.

Index (Termes fondamentaux)

Index

1. Liste des éléments contenus dans un fichier, assortie de clés ou de références destinées à localiser ses éléments.
2. Variable contenue dans un registre d'index pour permettre des comptages ou des modifications d'adresse.

Interactif (Termes fondamentaux)

Interactive

Mode assurant en temps réel le dialogue entre le programme utilisateur et les routines du système.

On dit aussi : **mode conversationnel**.

Interface (Matériel)

Interface

Circuit intégré assurant la jonction entre l'unité centrale et un périphérique.

Interpréteur (Logiciels de base)

Interpreter

Programme qui traduit en langage machine un programme écrit dans un autre langage de programmation, en exécutant immédiatement chaque instruction avant de passer à la traduction de la suivante.

Voir également : **Compilateur**.

Interpréteur de commandes (Logiciels de base)

Programme particulier dont le rôle est de permettre l'exécution de tous les programmes nécessaires à l'exploitation, aussi bien les programmes utilitaires que les programmes d'application.

K

Ko (Abréviation)

Kilo-octet. Vaut 2^{10} octets, soit 1024 octets.

Voir également : **octet**.

L

Langage (Programmation)

Language

Ensemble formé par un vocabulaire et ses règles de syntaxe, destiné à formuler sans ambiguïté des programmes.

Langage évolué (Programmation)

High level language

Langage de programmation qui n'est lié à la structure d'aucun ordinateur particulier ou d'aucune classe particulière d'ordinateurs, par opposition au langage d'assemblage (Voir **Assembleur**).

Langage machine (Programmation)

Machine language

Langage lié à l'ordinateur. Les instructions sont représentées par un code binaire directement assimilable par le micro-ordinateur.

Lecteur (Matériels)

Driver

Périphérique d'échange avec la mémoire centrale des données stockées sur des supports (disquettes, cassettes...).

Logiciel (Logiciels de base)

Software

Ensemble de programmes, relatif à un matériel, permettant le traitement de l'information.

Voir aussi : **Proiciel**.

Logiciel câblé (Logiciels de base)

Firmware

Ensemble de programmes résidants, enregistrés dans une mémoire non modifiable par l'utilisateur (ROM).

LOGO (Langages)

Langage de programmation destiné au traitement des listes et aux applications graphiques.

LSE (Langages)

Langage Symbolique d'Enseignement.

Langage de programmation à syntaxe française. LSE a permis de développer de nombreux logiciels éducatifs.

M

Matériel (Matériels)

Hardware

Terme générique qui comprend tout ou partie des éléments physiques qui composent le système informatique (par opposition à **logiciel**).

Mémoire (Termes fondamentaux)

Memory

Lieu de stockage des informations.

Mémoire centrale (Termes fondamentaux)

Main memory

Mémoire principale intérieure à l'unité centrale.

Mémoire de masse (Termes fondamentaux)

Mass storage

Mémoire extérieure au micro-ordinateur, permettant le stockage des fichiers.

On dit aussi : **mémoire externe**, **mémoire auxiliaire**.

Mémoire morte (Termes fondamentaux)

Read only memory

ROM = Read Only Memory ou **MEM**.

Mémoire réservée à la lecture. Son contenu est préservé en dehors de l'alimentation électrique.

Mémoire tampon (Termes fondamentaux)

Buffer

Mémoire spécialisée permettant, par stockage temporaire, le transfert de données entre deux organes ayant des caractéristiques de transfert différentes (organes non synchronisés, ou de débits inégaux).

Mémoire vive (Termes fondamentaux)

Random access memory

RAM = Random Access Memory ou **MEV**.

Mémoire permettant la lecture et l'écriture. Son contenu est effacé lorsque l'alimentation électrique n'est plus assurée.

On dit aussi : **mémoire volatile**.

Menu (Programmation)

Menu

Liste d'options offertes à l'utilisateur dans un programme interactif pour le guider dans ses choix. La sélection d'une option est faite à partir du clavier, d'une souris ou d'un crayon optique.

On dit aussi : **masque**.

Micro-ordinateur (Matériels)*Micro-computer*

Système basé sur un micro-processeur. Est caractérisé par son format réduit et son faible coût, en opposition aux moyens et aux gros systèmes informatiques.

On dit aussi : **ordinateur individuel, ordinateur personnel.**

Voir également : **ordinateur.**

Microprocesseur (Matériels)*Microprocessor*

Circuit intégré assurant l'essentiel des calculs et manipulations d'informations nécessaires au fonctionnement du micro-ordinateur.

Micro-programme (Logiciels)

Voir : **Logiciel câblé.**

Minidisque (Matériels)

Voir : **disquette.**

Mise au point (Programmation)*Debugging*

Détection, localisation et élimination des erreurs de programmation.

On dit aussi : **débogage.**

Mo (Abréviation)

Méga-octet. Vaut 2^{20} octets, soit 1 048 576 octets.

Voir également : **octet.**

Modem (Télécom)*Modem*

Modulateur - démodulateur.

Appareil servant à la conversion de signaux contenant l'information à chaque extrémité d'une ligne de transmission de données, par modulation à l'émission et démodulation à la réception. Transforme les données de l'ordinateur en signaux modulés transmissibles sur une ligne téléphonique.

Modules (Logiciels de base)*Modules*

Fichiers binaires indépendants stockés sur l'unité système et intégrés au système d'exploitation lors de la configuration logicielle (PRO-LOGUE).

On distingue les modules d'extension système et les modules d'extension environnement.

Moniteur

Monitor

1. (Matériels)
Ecran vidéo spécialisé dans la communication avec un ordinateur.
2. (Logiciels de base)
Programme stocké en mémoire morte permettant au microprocesseur de gérer la communication avec les périphériques dès la mise sous tension.

MS-DOS (Logiciels de base)

MicroSoft Disk Operating System.

Système d'exploitation très utilisé sur les micro-ordinateurs 16 bits.

Multi-critères (Logiciels de base)

Extension de l'accès séquentiel indexé. Cette méthode permet l'interrogation d'un fichier de données à partir d'une question à N critères portant sur une combinaison de rubriques. Le langage B.A.L., développé sous PROLOGUE, dispose d'instructions très performantes capables de gérer des fichiers multi-critères.

Voir également : **DIALOGUE.**

Multiplexeur (Télécom)

Multiplexor

Dispositif qui subdivise la capacité d'une ligne de transmissions de façon à la partager simultanément entre plusieurs utilisateurs :

- on parle de multiplexage temporel lorsque les données des différents utilisateurs sont imbriquées dans le temps.
- on parle de multiplexage en fréquence lorsque les données des différents utilisateurs sont transmises sur des bandes de fréquence différentes.

Multiposte (Logiciels de base)

Multi-user

Système d'exploitation qui peut gérer simultanément plusieurs postes utilisateurs rattachés à la même machine. Chaque poste se compose d'un écran et d'un clavier, permettant un dialogue opérateur/système. PROLOGUE est un système multiposte.

Multitâche (Logiciels de base)

Multitasking

Système d'exploitation qui autorise l'exécution simultanée de plusieurs tâches, suivant le principe du temps partagé. PROLOGUE est un système multitâche.

N

Numérique (Termes fondamentaux)

Digital

Qualifie une donnée représentée par un nombre entier ou fractionnaire ; s'oppose à **analogique**.

O

Octet (Termes fondamentaux)

Byte

Ensemble de 8 bits. Sert d'unité de capacité de mémoire.

Opérande (Programmation)

Operand

Donnée à laquelle s'applique une opération.

Opérateur (Programmation)

Operator

Symbole indiquant la nature d'une opération à effectuer.
Exemple : opérateur arithmétique.

Ordinateur (Matériels)

Computer

Dispositif conçu pour permettre le traitement automatisé des informations.

Tout ordinateur se compose d'une unité centralé, d'une unité arithmétique et logique et d'une unité de commande.

Voir également : **micro-ordinateur**.

Organigramme (Programmation)

Flowchart

Représentation graphique de l'enchaînement des opérations d'un programme.

On dit aussi : **ordinogramme**, **logigramme**.

P

Page (Termes fondamentaux)

Page

Partie logique de la mémoire tampon de l'écran sur laquelle on peut écrire et que l'on peut afficher.

Paramètre (Programmation)

Parameter

Voir : **attribut**.

Partition (Termes fondamentaux)

Partition

Zone de mémoire affectée à chaque chaîne de programmes.

PASCAL (Langages)

Langage de programmation structurée à vocation pédagogique.

Périphérique (Matériels)

Peripheral

Appareil relié à l'unité centrale et assurant la communication avec l'extérieur.

Le clavier est un périphérique d'entrée. L'écran est un périphérique de sortie.

Autres périphériques : lecteur de disquettes, imprimante, crayon optique, manette de jeux, etc.

Photostyle (Matériels)

Lightpen

Voir : **crayon optique**.

Pile (Termes fondamentaux)

Stack

Zone de mémoire accessible uniquement par son sommet : la dernière donnée entrée doit être la première sortie.

Piste (Termes fondamentaux)

Track

Sur une surface magnétique utilisée comme support, désigne l'ensemble des cellules de mémoire contiguës qui peuvent être desservies par une tête de lecture sans déplacement de celle-ci.

Pixel (Graphique)

Pixel

Désigne le plus petit élément de l'image vidéo. L'écran du Bull Micral 30 présente une grille de 320 × 200 soit 64 000 pixels.

Pointeur (Termes fondamentaux)

Pointer

Registre ou zone de mémoire destiné à recevoir l'adresse numérique d'un emplacement de mémoire : adresse d'une donnée, adresse de début d'une zone de mémoire.

Port (Matériels)

Port

Partie physique d'une interface de communication. Le nombre de ports sur une unité centrale détermine le nombre de canaux de communication pouvant être établi.

Porte (Termes fondamentaux)

Gate

Circuit logique possédant une seule sortie, dont l'état dépend exclusivement de l'état de ses entrées.

Prémarquage (Termes fondamentaux)

Voir : **formatage**.

Procédure (Termes fondamentaux)

Démarche suivie pour résoudre un problème.

Progiciel (Logiciels de base)

Package

Programme ou ensemble de programmes, conçu pour plusieurs utilisateurs en vue d'une application identique.

Programme (Programmation)

Routine

Suite d'instructions et d'indications, rédigées dans un langage donné, compréhensibles par un ordinateur et correspondant à la formulation d'un traitement.

PROLOGUE (Logiciels de base)

Système d'exploitation développé par Bull Micral. Disponible sur micro-ordinateurs 8 et 16 bits, il peut gérer des systèmes multi-tâches et multi-postes. Associé au langage B.A.L., il permet la création de structures de données évoluées (séquentiel indexé, multi-critères, bases de données) et offre de nombreuses possibilités de télécommunications.

Protocole (Télécom)

Ensemble de méthodes assurant une transmission correcte des données. Définit notamment le code, les caractères et le mode de transmission utilisés.

R

Recopie d'écran (Termes fondamentaux)

Hard copy

Opération qui consiste à reproduire sur papier la page-écran présente sur la console.

Recouvrement (Termes fondamentaux)

Overlay

Occupation successive d'une même zone de mémoire par différentes parties d'un programme.

Registre (Termes fondamentaux)

Register

Mémoire spécialisée ou zone de mémoire, de capacité limitée, affectée à des fonctions particulières pour concourir à l'exécution des opérations dans un traitement.

Relationnel (Logiciels de base)

Relational

Voir : **S.G.B.D.**

Répertoire (Logiciels de base)

Directory

Liste des noms et des adresses des fichiers d'une disquette, généralement accessible par une commande ou un utilitaire du système d'exploitation.

On dit aussi : **catalogue**.

Répertoire hiérarchisé (Logiciels de base)

Pour une gestion plus simple des fichiers, MS-DOS autorise la création de répertoires supplémentaires, appelés sous-répertoires. Ceux-ci sont indépendants les uns des autres.

C'est cette structure (semblable à un arbre, dans laquelle le répertoire principal créé par MS-DOS constitue la racine) que l'on appelle structure de répertoires hiérarchisés.

Réseau (Télécom)

Network

Ensemble de sites émetteurs et récepteurs d'informations reliés entre eux par des lignes de transmission.

Réseau local (Télécom)

Local network

Réseau de communication dont les sites et les lignes sont répartis sur un domaine limité (bâtiment, entreprise, école,...).

Résolution (Graphique)*Définition*

Désigne le nombre de points identifiables sur un écran.

La précision des images ou des caractères reproduits sur un écran est proportionnelle à la densité de points. On parle de haute et de basse résolutions.

Routine (Programmation)*Routine*

Tout ou partie d'un programme ayant un emploi général ou répété, souvent inclus dans la bibliothèque.

Voir également : **sous-programme**.

S**Sauvegarde** (Termes fondamentaux)*Back up*

Opération consistant à recopier des informations déjà stockées sur une mémoire externe (disquette, disque dur) sur un autre support pour des raisons de sécurité.

Voir également : **dérouleuse de bande**.

Secteur (Termes Fondamentaux)*Sector*

Partie de la piste d'une surface magnétique telle que la piste contienne un nombre entier de secteurs, chaque secteur ayant une adresse dans la piste.

Segment*Segment*

1. (Termes fondamentaux)

Découpage de la mémoire.

2. (Programmation)

En B.A.L., section autonome de lignes d'instructions dans un programme. Ce découpage permet une gestion dynamique de la mémoire, car chaque segment est chargé et exécuté selon son numéro. Le chaînage entre segments reste à la charge du programmeur.

Segmentation (Termes fondamentaux)

Division d'un programme en plusieurs segments liés à différents langages évolués (B.A.L.).

Série (Télécom)*Serial*

Qualifie une transmission dans laquelle les éléments constitutifs des données sont transmis l'un après l'autre. S'oppose à **parallèle**.

Serveur (Télécom)*Server*

Ordinateur central stockant les informations consultées à distance par les terminaux utilisateurs dans un réseau local.

S.G.B.D. (Bases de données)*Data Base Management System (DBMS)***1. Système de Gestion de Bases de Données.**

On distingue à l'heure actuelle trois familles de S.G.B.D. Chaque famille se réfère à un modèle de données très particulier :

- le modèle hiérarchique.
- le modèle réseau.
- le modèle relationnel.

2. Le modèle « Bases de données » du système d'exploitation PROLOGUE est de type relationnel. L'accès à la base peut s'effectuer de deux façons :

- soit par un programme B.A.L.
- soit par DIALOGUE.

Modèle relationnel :

Il consiste à définir chaque donnée élémentaire par son nom et l'ensemble des valeurs qu'elle peut prendre. Les données sont liées entre elles par des relations qui définissent des sous-ensembles au sein de la base.

Simulateur (Logiciels de base)*Simulator*

Programme spécifique d'un type d'ordinateur, destiné à transformer dans son langage propre et à adapter à son organisation interne, un programme rédigé dans le langage d'un autre type d'ordinateur.

Voir également : **émulateur** = dispositif permettant d'obtenir le même résultat.

Sortie (Termes fondamentaux)*Output*

Emission de données internes par un ordinateur ou l'un de ses éléments.

Sous-programme (Programmation)*Subroutine*

Séquence d'instructions, isolée et individualisée, généralement utilisée en plusieurs points d'un même programme ou dans d'autres programmes.

Souris (Matériels)*Mouse*

Périphérique constitué d'un boîtier relié à l'ordinateur, dont le mouvement entraîne un déplacement du curseur sur l'écran. Permet ainsi de désigner des parties d'écran ou de choisir des options d'un menu qui s'exécutent alors sans avoir recours au clavier.

Support (Matériels)*Medium*

Tout matériel destiné à recevoir l'enregistrement de données.

Synchrone (Télécom)*Synchronous*

Qualifie un mode de transmission de données dans lequel l'intervalle de temps entre le début de la transmission d'un caractère et le début de la transmission du suivant, est constant.

Système d'exploitation (Logiciels de base)*Operating system*

Logiciel indépendant des applications et indispensable au fonctionnement de l'ordinateur. Il planifie l'enchaînement des tâches, gère les fichiers, et contient souvent des utilitaires. Les principaux systèmes d'exploitation sur micro-ordinateurs sont : **CP/M, MS-DOS, et PROLOGUE.**

T

Téléchargement (Télécom)

Teleloading

Envoi, par un procédé télématique, d'un programme à exécuter ou à mémoriser par l'appareil récepteur.

Téléinformatique (Télécom)

Teleprocessing, Data communication

Ensemble des moyens logiciels et matériels qui permettent le traitement des informations à distance.

Télématique (Télécom)

Acronyme de **télécommunications** et d'**informatique**. Echanges d'informations à travers le réseau téléphonique et plus largement des télécommunications.

Temps partagé (Termes fondamentaux)

Time sharing

Mode de traitement dans lequel les données sont gérées par séquences disponibles successives.

Temps réel (Termes fondamentaux)

Real time

Mode de traitement qui prend directement en compte les données au fur et à mesure de leur arrivée.

Terminal (Matériels)

Terminal unit

Périphérique ou ensemble de périphériques d'un ordinateur relié à l'unité centrale par une ligne de transmission de données.

Test (Termes fondamentaux)

Test

Opération consistant à reconnaître une situation ou une condition.

Traducteur (Logiciels de base)

Translator

Programme qui transforme un programme énoncé dans un langage de programmation de manière à l'exprimer dans un autre langage de programmation.

Trame

Frame

1. (Télécom)

Séquence de bits incluant les données et une ou plusieurs adresses.

2. (Termes fondamentaux)

Ensemble d'éléments d'information et d'éléments de contrôle organisé suivant une structure définie.

Transistor (Matériels)*Transistor*

Composant électronique jouant le rôle d'un interrupteur ou d'un amplificateur.

U**Unidirectionnel (Télécom)***Simplex*

Qualifie une transmission qui ne peut être effectuée que dans un sens.

Unité centrale (Matériels)*Central Process Unit (CPU)*

Désigne l'ensemble matériel où réside le processeur, par opposition aux périphériques.

Utilitaire (Logiciels de base)*Utility program*

Programme généralement inclus dans le système d'exploitation, effectuant des opérations systématiques de routine telles que création et copie de fichiers.

V**Volume (Termes fondamentaux)***Volume*

Unité de stockage utilisée pour l'organisation logique des fichiers. Le système PROLOGUE considère chaque unité de disque comme un volume unique défini par son nom et ses caractéristiques de stockage.

Bull



VOS REMARQUES SUR CE DOCUMENT / TECHNICAL PUBLICATIONS REMARKS FORM

TITRE / TITLE

PROLOGUE

Guide de l'Utilisateur

N° REFERENCE / REFERENCE No :

05 F2 9663 REV4

DATE / DATED :

Avril 1987

ERREURS DETECTÉES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGERÉES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

► Vos remarques et suggestions seront attentivement examinées.

► *Your comments will be promptly investigated by qualified technical personnel
and action will be taken as required.*

NOM / NAME : DATE

SOCIÉTÉ / COMPANY :

ADRESSE / ADDRESS :

■ Envoyez ce document à :

**BULL MTS
Documentation BULL MICRAL**

■ Send this sheet to :

**7, rue Ampère
B.P. 92
91301 MASSY CEDEX - FRANCE**

Rédigé pour BULL MICRAL par Monsieur J.C. COULON

1991



7, rue Ampère
B.P. 92
91301 MASSY CEDEX - FRANCE
TELEX 690079F



05F29663 04

